

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту

Завідувач кафедри _____ О.О. Ємець
(підпис)

«_____» _____ 2019 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОЇ РОБОТИ**

**на тему
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТРЕНАЖЕРУ З ТЕМИ «ПРАВИЛА
ВИВЕДЕННЯ» ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ
«МАТЕМАТИЧНА ЛОГІКА»**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Стовбун Дмитро Олегович

_____ «___» _____ 2019р.
(підпис)

Науковий керівник к.ф.-м.н., доц. Черненко Оксана Олексіївна

_____ «___» _____ 2019р.
(підпис)

ПОЛТАВА 2019р.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	3
ВСТУП	4
1. ПОСТАНОВКА ЗАДАЧІ.....	6
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	8
2.1. Огляд тренажерів з математичних дисциплін	8
2.2. Позитивні аспекти оглянутих тренажерів	14
2.3. Негативні аспекти оглянутих тренажерів.....	14
2.4. Актуальність теми роботи.....	15
3. ТЕОРЕТИЧНА ЧАСТИНА	17
3.1. Формальна теорія	17
3.2. Приклади застосування правил виведення.....	20
3.3. Алгоритмізація за темою роботи.....	23
3.4. Розробка блок-схеми.....	36
4. ПРАКТИЧНА ЧАСТИНА	43
4.1. Обґрунтування вибору програмних засобів.....	43
4.2. Опис процесу програмної реалізації	45
4.3. Опис програми.....	51
4.4. Інструкція по використанню тренажеру.....	55
ВИСНОВКИ.....	60
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	63
ДОДАТОК А. КОД ПРОГРАМИ	Ошибка! Закладка не определена.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
тренажер	комп'ютерна програма, призначена для вивчення і закріплення різноманітних практичних навичок
NetBeans IDE	середовище розробки
Java	об'єктно-орієнтована мова програмування
Формальна система	сукупність абстрактних об'єктів, не пов'язаних із зовнішнім світом, в якій представлено правила оперування множиною символів у строго синтаксичному трактуванні без врахування смислового змісту, тобто семантики
Правила виведення	обґрунтовують кроки доведення логічних теорем, яке полягає в перевірці того, що висновок являє собою логічний наслідок множини гіпотез

ВСТУП

Сучасний рівень розвитку комп'ютерної техніки і програмного забезпечення надає широкі можливості щодо модернізації та підвищення ефективності навчання. Використання мультимедійних технологій у навчальному процесі урізноманітнює і автоматизує його, підвищує ефективність засвоєння матеріалу, здійснює контроль знань. У зв'язку з цим можна стверджувати, що впровадження інформаційних технологій у повсякденне життя українських громадян стало поштовхом для розвитку процесів пов'язаних з дистанційним навчанням.

Навчаючись на дистанційних курсах студент підвищує свій інтелектуальний і творчий потенціал, навчається самостійно приймати рішення.

Метою роботи є розробка елементів програмного забезпечення тренажеру з теми «Правила виведення» дистанційного навчального курсу «Математична логіка» та закріплення знань і навичок.

Об'єктом розробки в даній роботі є процес дистанційного навчання математичним дисциплінам.

Предметом розробки є програмний продукт, що реалізує тренажер для закріплення знань із застосування правил виведення в численні висловлювань.

Головне завдання – розробити алгоритм роботи тренажеру по якому вже буде розроблено навчальний тренажер з теми «Правила виведення».

Методи розробки – правила виведення в численні висловлювань і їх застосування.

При реалізації тренажера використано середовище візуальної розробки програм NetBeans IDE та об'єктно-орієнтована мова програмування Java.

В програмі реалізовано можливість переключення між українською та англійською мовами.

Тренажер готовий до використання в дистанційному курсі «Математична логіка».

Робота складається з чотирьох розділів. У першому розділі розглянуто постановку задачі для реалізації тренажера. У другому розділі описано огляд тренажерів з математичних дисциплін, їх позитивні та негативні аспекти, актуальність теми роботи. У третьому розділі представлено теоретичний матеріал по правилам виведення, наведено приклади застосування та описано алгоритм роботи тренажеру, представлено блок-схему алгоритму. У четвертому розділі – описано обґрунтування вибору програмних засобів, процес програмної реалізації, саму програму і інструкцію по використанню тренажера.

Обсяг пояснювальної записки: 77 стор., в т.ч. основна частина - 55 стор., джерела - 15 назв.

1. ПОСТАНОВКА ЗАДАЧІ

Першим кроком потрібно скласти алгоритм роботи тренажеру по якому вже буде розроблено навчальний тренажер, як складову дистанційного курсу «Математична логіка».

Розглянемо основні завдання роботи:

- розглянути тренажери схожої тематики;
- розглянути тренажери математичних дисциплін з інших тематик;
- визначити позитивні і негативні аспекти оглянутих тренажерів;
- описати актуальність теми роботи;
- розглянути теоретичні відомості про правила виведення в численні висловлювань;
- розглянути специфіку застосування правил виведення в численні висловлювань;
- розробити алгоритм тренажеру, що дозволить закріпити знання з теми «Правила виведення»;
- розробити навчальний тренажер з даної теми.

Перед проходженням тренажеру слід розробити можливість звернутися до теоретичного матеріалу і прикладів. В алгоритмі слід покроково описати всі дії при розв'язанні прикладів, де необхідно застосувати правила виведення в численні висловлювань.

Також у тренажері необхідно реалізувати декілька прикладів, що ознайомлять користувача з самими правилами виведення і їх застосуванням. При завершенні проходження слід розробити можливість перейти до наступного прикладу або розпочати спочатку його проходження при розгляді всіх прикладів [2].

При невірній відповіді потрібно виводити повідомлення про помилку, яке міститиме вірну відповідь.

Оскільки тренажер може також використовуватися у навчальному процесі іноземних студентів, то необхідно реалізувати можливість переключення між українською та англійською мовами. Теоретичний матеріал повинен бути теж в двох варіантах. При початку проходження блокується можливість вибору мови.

Результати роботи мають бути опубліковані в збірнику наукових статей та матеріалах науково-практичного семінару.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Огляд тренажерів з математичних дисциплін

Було розглянуто декілька тренажерів з дисципліни «Методи оптимізації та дослідження операцій». В тренажері з теми «Угорський метод в задачі про призначення» на стартовій сторінці виводиться його назва, а також вказується автор-розробник, керівник і тема (рис. 2.1) [3-5].

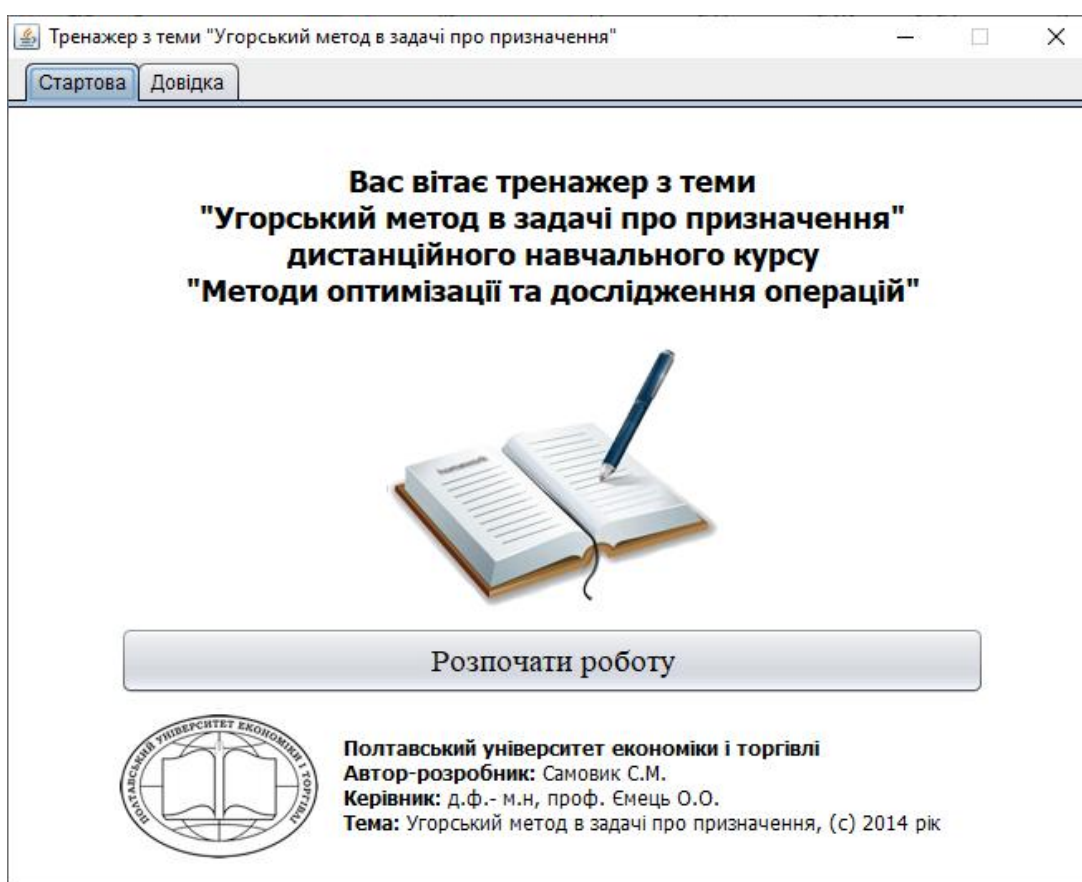


Рисунок 2.1 – Стартова сторінка тренажеру з теми «Угорський метод в задачі про призначення»

Відразу можна перейти до довідки, або зробити це під час проходження тренажеру. (рис. 2.2). Розпочавши тренажер є можливість повернутися на стартову або почати виконання задачі (рис. 2.3).

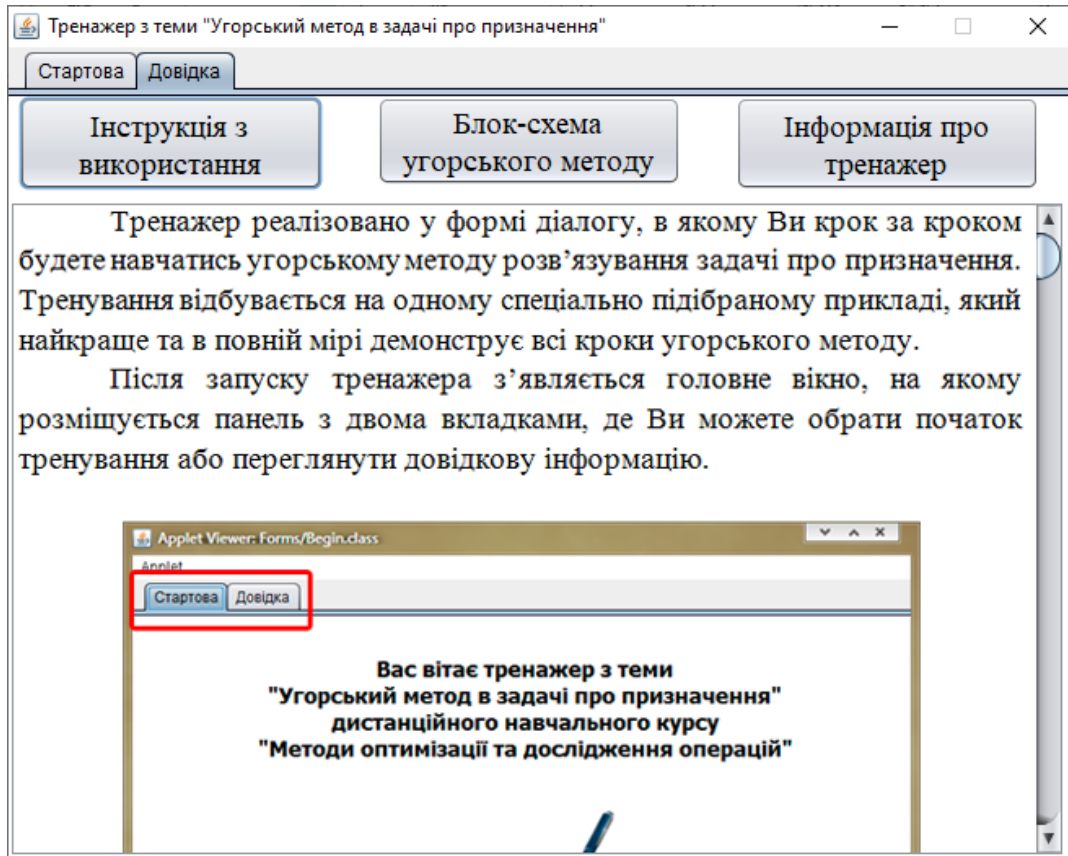


Рисунок 2.2 – Довідка тренажеру

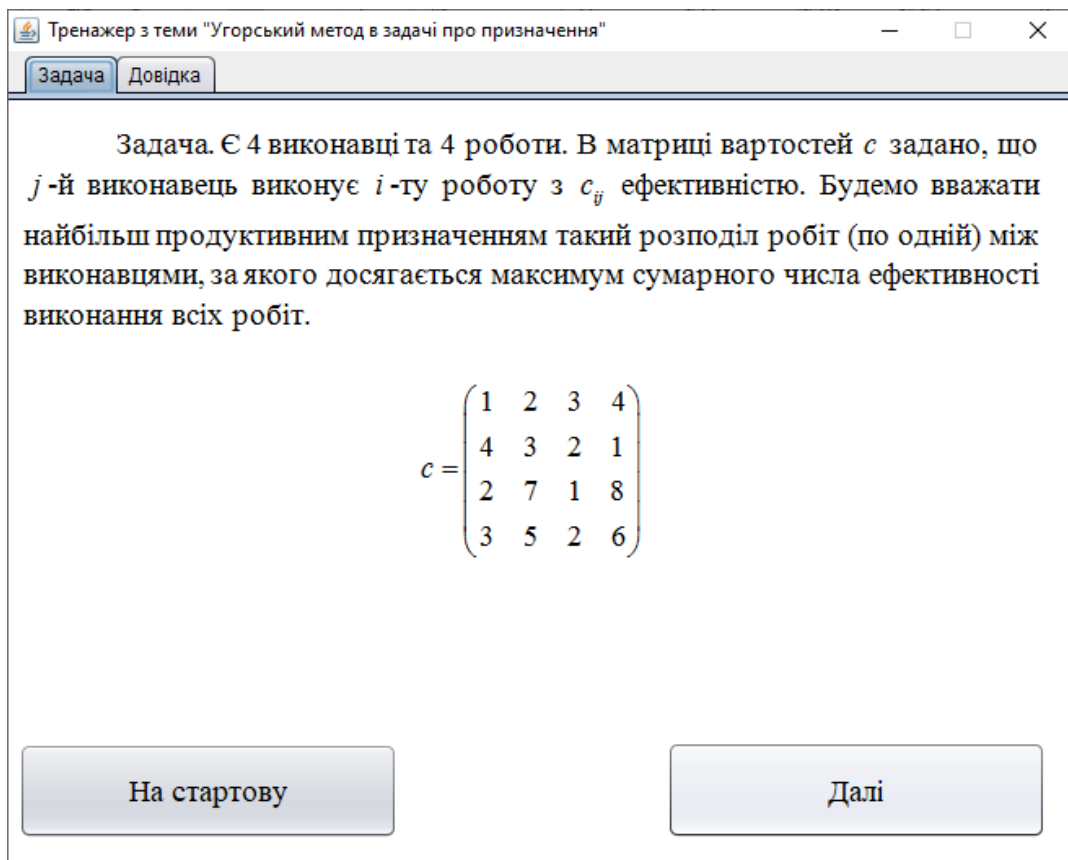


Рисунок 2.3 – Умова задачі

В тренажері реалізовано декілька типів питань, перший – це завдання з вибором відповіді (рис. 2.4).

Тренажер з теми "Угорський метод в задачі про призначення"

Тренування Довідка

Що потрібно зробити на першому кроці алгоритму?

☒ А. Знайти мінімальні елементи в рядках.

☐ Б. Від максимальних елементів відняти елементи стовпців.

☐ В. Відняти мінімальні елементи від елементів рядків.

☐ Г. Знайти максимальні елементи в стовпцях.

Далі

Рисунок 2.4 – Умова задачі

Інший тип завдань – це заповнення комірок або таблиці (рис. 2.5, 2.6).

Тренажер з теми "Угорський метод в задачі про призначення"

Тренування Довідка

Визначте в кожному із стовпців максимальний елемент і запишіть його до відповідної клітинки.

1	2	3	4
4	3	2	1
2	7	1	8
3	5	2	6

1 2 3 4

Далі

Рисунок 2.5 – Заповнення комірок

Тренажер з теми "Угорський метод в задачі про призначення"

Тренування Довідка

Від знайдених на попередньому кроці максимальних елементів відніміть елементи відповідних стовпців і запишіть до таблиці.

Дані з попереднього кроку

1	2	3	4
4	3	2	1
2	7	1	8
3	5	2	6
4	7	3	8

3	5	0	4
0	4	1	7
2	0	2	0
1	2	1	2

Далі

Рисунок 2.6 – Заповнення таблиці

У разі неправильної відповіді виводиться відповідне повідомлення (рис. 2.7).

Тренажер з теми "Угорський метод в задачі про призначення"

Тренування Довідка

ПОМИЛКА!!!

Запам'ятайте! На першому кроці знаходяться максимальні елементи в стовпцях.

Далі

Рисунок 2.7 – Помилка

Наступний тренажер ознайомлює користувача з графічним методом розв'язування задач лінійного програмування (рис. 2.8) [6-7].

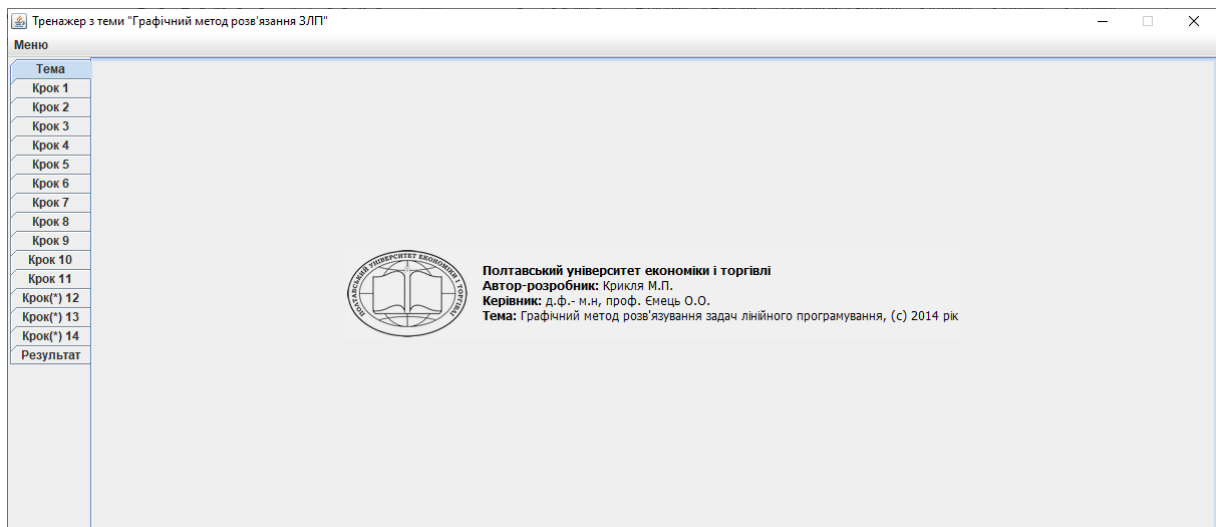


Рисунок 2.8 – Стартова сторінка тренажеру з теми «Графічний метод розв'язування задач лінійного програмування»

Більшість кроків зроблені з вибором одного або декількох варіантів відповідей (рис. 2.9). Також наявна побудова графіка (рис. 2.10).

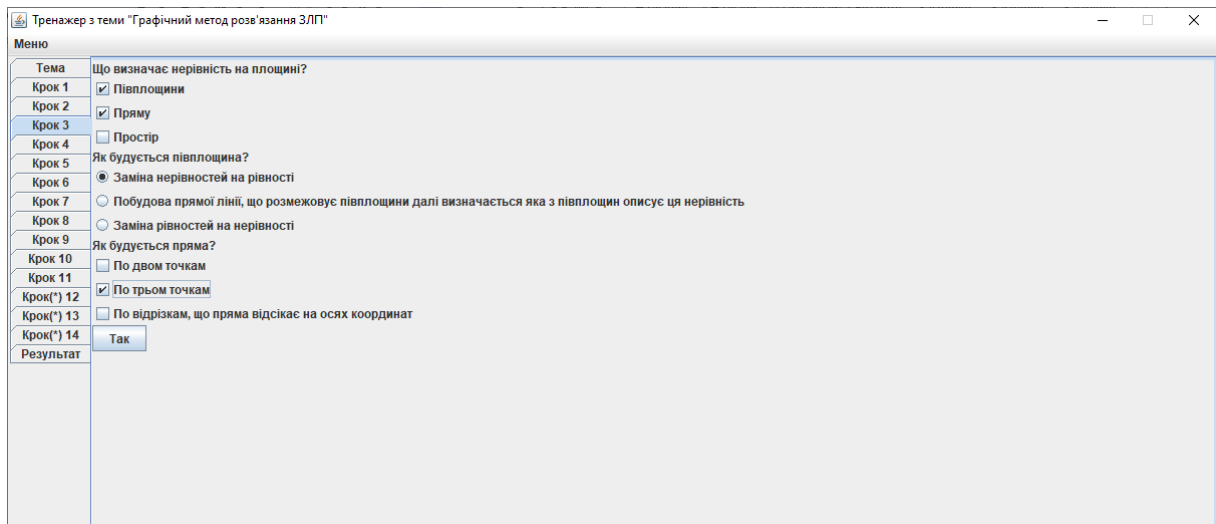


Рисунок 2.9 – Вибір одного або декількох варіантів відповідей

У разі неправильної відповіді відображається повідомлення про помилку (рис. 2.10). Серед кроків також наявна побудова графіка (рис. 2.11).

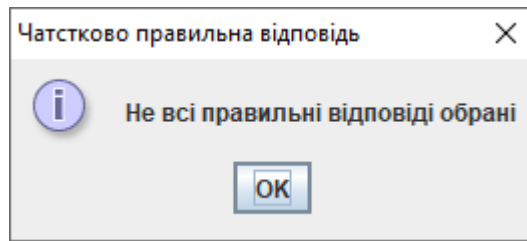


Рисунок 2.10 – Помилка



Рисунок 2.11 – Побудова графіка

Останнім розглянуто тренажер з теми «Двоїстість в лінійному програмуванні» (рис. 2.12) [7-8].

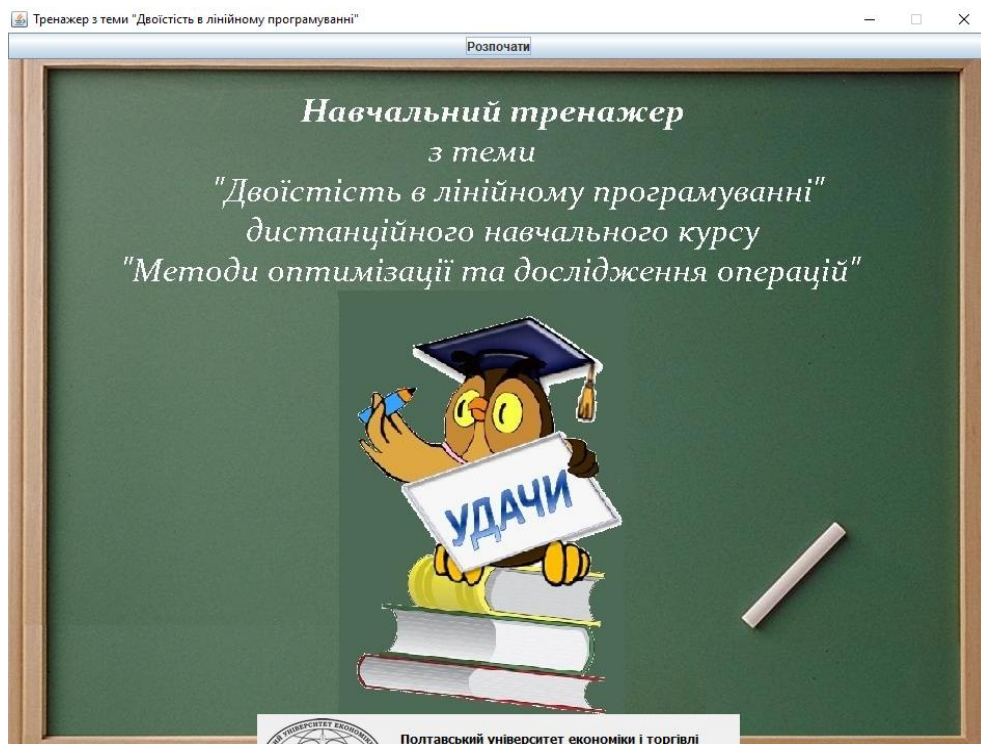


Рисунок 2.12 – Стартова сторінка тренажеру з теми «Двоїстість в лінійному програмуванні»

В ньому більшість питань з вибором однієї відповіді, при помилці виводиться відповідне повідомлення (рис. 2.13).

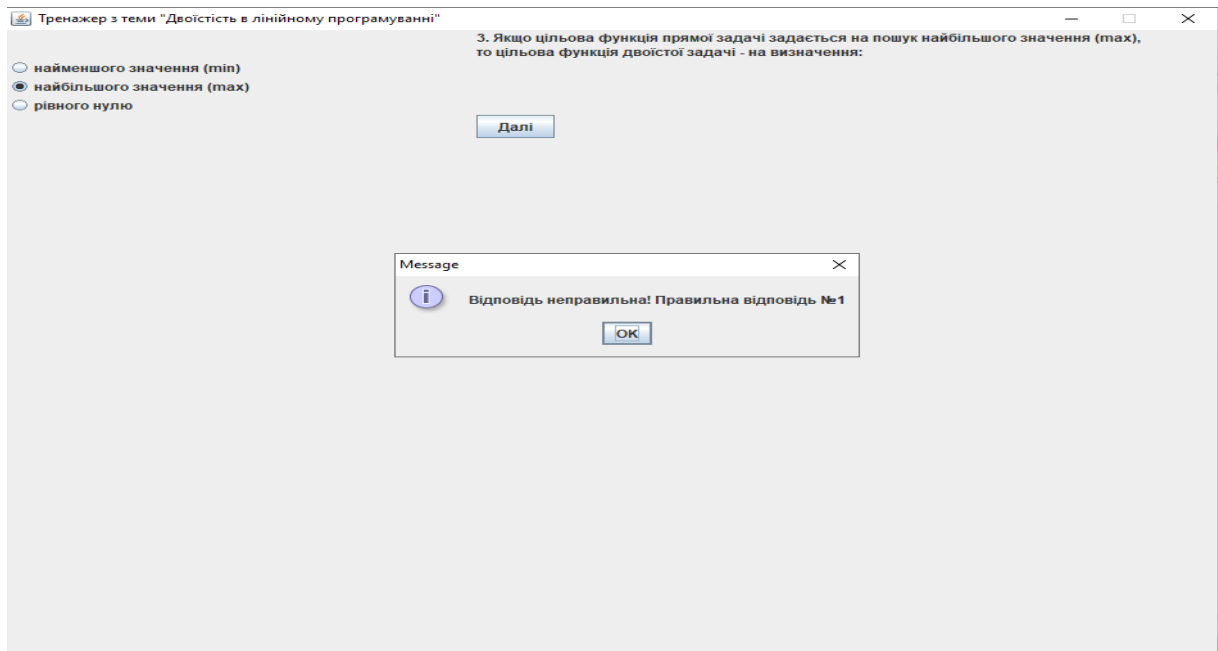


Рисунок 2.13 – Вибір відповіді та повідомлення про помилку

2.2. Позитивні аспекти оглянутих тренажерів

До позитивних аспектів оглянутих тренажерів можна віднести:

- зручний і зрозумілий дизайн;
- наявність довідки;
- миттєвий доступ до довідки в будь-який момент;
- поступовий перехід по кроках для закріплення матеріалу;
- виведення повідомлення про помилку при невірній відповіді;
- інформативність помилки;
- відображення результату проходження тренажеру.

2.3. Негативні аспекти оглянутих тренажерів

Серед негативних аспектів оглянутих тренажерів можна зазначити:

- незрозумілий інтерфейс програми;

- не відразу зрозуміло що потрібно зробити на певному кроці;
- не завжди зрозуміло що саме слід вводити у конкретні поля;
- забагато пустого місця в тренажері;
- між кроками необхідно переключатися самому;
- при неправильній відповіді відображається повідомлення із незрозумілою підказкою;
- некоректна робота деяких елементів тренажеру.

2.4. Актуальність теми роботи

Специфіка організації навчального процесу на підготовчому факультеті, потреба у підвищенні ефективності освітнього процесу, реалізація принципів індивідуального підходу в навчанні змушують відводити особливе місце самостійній роботі студентів (СРС) як активній пізнавальній діяльності тих, хто навчається.

Традиційно СРС розмежовують на аудиторну і позааудиторну. Крім фактичного місця проведення СРС, що дало назву цим видам, вони характеризуються рядом специфічних ознак. Так, аудиторна СРС характеризується взаємодією викладача і студента в процесі безпосереднього спілкування, чітким розподілом функцій між викладачем і студентом, наявністю певного часового режиму. Позааудиторна СРС відрізняється опосередкованим і обмеженим керівництвом викладача, індивідуальністю навчальної діяльності, квазідетермінованими витратами часу.

Інтенсивний розвиток інформаційно-комунікаційних технологій, який підняв сучасні технічні засоби навчання на новий рівень оснащення і можливостей, дозволяє переглянути основні підходи до організації аудиторної та позааудиторної СРС з метою підвищення ефективності освітнього процесу.

Таким чином, актуальним є завдання створення сучасних технічних засобів навчання як комплексу апаратного забезпечення, програмного забезпечення і електронних дидактичних засобів з метою розширення можливостей та удосконалення організації аудиторної і позааудиторної СРС.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Формальна теорія

Формальна система — результат строгої формалізації теорії, яка передбачає повну абстракцію від сенсу слів використовуваної мови, причому всі умови, що регулюють вживання цих слів в теорії, явно висловлені за допомогою аксіом і правил, що дозволяють вивести одну фразу з інших.

Формальна система — це сукупність абстрактних об'єктів, не пов'язаних із зовнішнім світом, в якій представлено правила оперування множиною символів у строго синтаксичному трактуванні без врахування смислового змісту, тобто семантики.

Строго описані формальні системи з'явилися після того, як було поставлено задачу розв'язності Гільберта. Перші ФС з'явилися після виходу книг Рассела та Уайтхеда «Формальні системи». Цим формальним системам було пред'явлено певні вимоги.

Формальна теорія вважається визначеною, якщо:

1. Задано скінченну або зліченну множину довільних символів. Скінченні послідовності символів називаються виразами теорії.
2. Є підмножина виразів, званих формулами.
3. Виділено підмножину формул, званих аксіомами.
4. Є скінченна множина відношень між формулами, званих правилами виведення.

Зазвичай є ефективна процедура, що дозволяє за даним виразом визначити, чи є він формулою. Часто множина формул задається індуктивним визначенням. Як правило, ця множина є нескінченною. Множина символів і множина формул у сукупності визначають мову або сигнатуру формальної теорії.

Найчастіше мається можливість ефективно з'ясовувати, чи є дана формула аксіомою; в такому випадку теорія називається ефективно аксіоматизованою або аксіоматичною. Множина аксіом може бути скінченною або нескінченною. Якщо кількість аксіом скінченна, то теорія називається скінчено аксіоматизованою. Якщо множина аксіом нескінченна, то, як правило, вона задається за допомогою скінченного числа схем аксіом і правил породження конкретних аксіом зі схеми аксіом. Зазвичай аксіоми поділяються на два види: логічні аксіоми (спільні для цілого класу формальних теорій) і нелогічні або власні аксіоми (визначають специфіку та зміст конкретної теорії).

Для кожного правила виведення R і для кожної формули A ефективно розв'язується питання про те, чи знаходиться обраний набір формул у відношенні R з формулою A , і якщо так, то A називається безпосереднім наслідком даних формул за правилом R . Виведенням називається всяка послідовність формул така, що всяка формула послідовності є або аксіомою, або безпосереднім наслідком якихось попередніх формул за одним з правил виведення.

Формула називається теоремою, якщо існує виведення, в якому ця формула є останньою.

Теорія, для якої існує ефективний алгоритм, що дозволяє дізнаватися по даній формулі, чи існує її виведення, називається розв'язною; в іншому випадку теорія називається нерозв'язною.

Теорія, в якій не всі формули є теоремами, називається абсолютно несуперечливою.

Дедуктивна теорія вважається заданою, якщо:

- Задано алфавіт (множину) і правила утворення виразів (слів) у цьому алфавіті.
- Задано правила утворення формул (правильно побудованих, коректних висловів).

- З множини формул деяким способом виділено підмножину T теорем (доказових формул).

Різновиди дедуктивних теорій залежно від способу побудови множини теорем:

1. Вказання аксіом та правил виведення.

У множині формул виділяється підмножина аксіом, і задається скінченне число правил виведення — таких правил, за допомогою яких (і тільки за допомогою їх) з аксіом і раніше виведених теорем можна утворити нові теореми. Всі аксіоми також входять до числа теорем. Іноді (наприклад, в аксіоматиці Пеано) теорія містить нескінченну кількість аксіом, що задаються за допомогою однієї або декількох схем аксіом. Аксіоми іноді називають «прихованими визначеннями». Таким способом задається формальна теорія (формальна аксіоматична теорія, формальне (логічне) числення).

2. Вказання лише аксіом

Задаються лише аксіоми, правила виведення вважаються загальновідомими.

При такому заданні теорем кажуть, що задано напівформальну аксіоматичну теорію.

3. Вказання лише правил виведення

Аксіом немає (множина аксіом порожня), задаються лише правила виведення. По суті, задана таким чином теорія — окремий випадок формальної, але має власну назву: теорія природного виведення.

Властивості дедуктивних теорій:

1. Несуперечність.

Теорія, в якій множина теорем покриває всі множини формул (всі формули є теоремами, «істинними висловлюваннями»), називається суперечливою. В іншому випадку теорія називається несуперечливою. З'ясування суперечливості теорії — одне з найважливіших й іноді найскладніших задач формальної логіки. Після з'ясування суперечливості

теорія, як правило, не має подальшого ані теоретичного, ані практичного застосування.

2. Повнота.

Теорія називається повною, якщо в ній для будь-якої формули F виводиться або сама F , або її заперечення $\neg F$. В іншому випадку, теорія містить недоведені твердження (твердження, які не можна ані довести, ані спростувати засобами самої теорії), і називається неповною.

Формальна аксіоматична теорія числення висловлень є повною відносно своєї моделі алгебри висловлень.

3. Незалежність аксіом.

Окрема аксіома теорії вважається незалежною, якщо цю аксіому не можна вивести з інших аксіом. Залежна аксіома по суті є надмірною, і її видалення з системи аксіом ніяк не відіб'ється на теорії. Вся система аксіом теорії називається незалежною, якщо кожна аксіома в ній незалежна.

4. Розв'язність.

Теорія називається розв'язною, якщо в ній поняття теореми ефективне, тобто існує ефективний процес (алгоритм), що дозволяє для будь-якої формули за зліченне число кроків визначити, є вона теоремою чи ні [9-10].

3.2. Приклади застосування правил виведення

1. Правила виведення в численні висловлювань.

Правила виведення обґрунтовують кроки доведення логічних теорем, яке полягає в перевірці того, що висновок являє собою логічний наслідок множини гіпотез. Найважливіші правила виведення та відповідні їм тавтології наведено в таблиці 3.1.

Таблиця 3.1 – Правила виведення

Правило виведення	Тавтологія	Назва правила
$p \mid -p \vee q$	$p \rightarrow (p \vee q)$	Уведення диз'юнкції
$p \wedge q \mid -p$	$(p \wedge q) \rightarrow p$	Виключення кон'юнкції
$p, q \mid -p \wedge q$	$((p) \wedge (q)) \rightarrow (p \wedge q)$	Уведення кон'юнкції
$p, p \rightarrow q \mid -q$	$(p \wedge (p \rightarrow q)) \rightarrow q$	Modus ponens
$\bar{q}, p \rightarrow q \mid -\bar{p}$	$(\bar{q} \wedge (p \rightarrow q)) \rightarrow \bar{p}$	Modus tollens
$p \rightarrow q, q \rightarrow r \mid -p \rightarrow r$	$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow \rightarrow (p \rightarrow r)$	Гіпотетичний силогізм
$p \vee q, \bar{p} \mid -q$	$((p \vee q) \wedge \bar{p}) \rightarrow q$	Диз'юнктивний силогізм
$p \vee q, \bar{p} \vee r \mid -q \vee r$	$((p \vee q) \wedge (\bar{p} \vee r)) \rightarrow \rightarrow (q \vee r)$	Резолюція

2. Застосування правил виведення в численні висловлювань.

Приклад 1.

Які правила виведення використані в міркуваннях:

- 1) Похолоднішало. Отже, похолоднішало чи почав падати дощ.
- 2) Похолоднішало та почав падати дощ. Отже, похолоднішало.
- 3) Якщо сьогодні буде дощ, то сьогодні ми не поїдемо на пікнік.

Якщо ми не поїдемо на пікнік сьогодні, то поїдемо на пікнік завтра. Отже, якщо сьогодні буде дощ, то ми поїдемо на пікнік завтра.

Розв'язання.

Нехай p – це висловлювання «Похолоднішало», а q – висловлювання «Почав падати дощ».

Тоді:

- твердження 1) можна записати у вигляді правила введення диз'юнкції $p \mid -p \vee q$;

- твердження 2) у вигляді правила виключення кон'юнкції $p \wedge q | - p$.

- для твердження 3) нехай p : «Сьогодні буде дощ», q : «Сьогодні ми не поїдемо на пікнік», а r : «Ми поїдемо на пікнік завтра». Тоді твердження можна записати у вигляді правила $((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$ гіпотетичного силогізму.

Приклад 2.

Довести, що з гіпотез «Сьогодні не сонячний день і холодніше, ніж учора», «Ми підемо купатися лише якщо сьогодні сонячний день», «Якщо ми не підемо купатись, то поїдемо плавати на човні» та «Якщо ми поїдемо плавати на човні, то повернемося пізно ввечері» впливає висновок «Ми повернемося пізно ввечері».

Розв'язання.

Нехай

- p : «Сьогодні сонячний день»;
- q : «Сьогодні холодніше, ніж учора»;
- r : «Ми підемо купатися»;
- s : «Ми поїдемо плавати на човні»;
- t : «Ми повернемося пізно ввечері».

Тут гіпотези – $\bar{p} \wedge q$, $r \rightarrow p$, $\bar{r} \rightarrow s$, $s \rightarrow t$, а висновок – t . Наведемо послідовність з 8 кроків отримання висновку із заданої множини гіпотез і зазначимо застосовані правила виведення:

- 1) $\bar{p} \wedge q$ – гіпотеза;
- 2) \bar{p} – правило виключення кон'юнкції до 1;
- 3) $r \rightarrow p$ – гіпотеза;
- 4) \bar{r} – modus tollens до 2 та 3;
- 5) $\bar{r} \rightarrow s$ – гіпотеза;
- 6) s – modus ponens до 4 та 5;
- 7) $s \rightarrow t$ – гіпотеза;

8) t – modus ponens до 6 та 7.

Приклад 3.

Довести, що з гіпотез «Якщо ти надішлеш мені повідомлення електронною поштою, то я закінчу писати програму», «Якщо ти не надішлеш мені повідомлення електронною поштою, то я рано піду спати» та «Якщо я рано піду спати, то прокинуся бадьорим» впливає висновок «Якщо я не закінчу писати програму, то я прокинуся бадьорим».

Розв'язання.

Уведемо такі позначення:

- p : «Ти надішлеш мені повідомлення електронною поштою»,
- q : «Я закінчу писати програму»,
- r : «Я рано піду спати»,
- s : «Я прокинуся бадьорим».

Гіпотези можна записати у вигляді, $p \rightarrow q$, $\bar{p} \rightarrow r$, $r \rightarrow s$. Потрібно обґрунтувати висновок $\bar{q} \rightarrow s$. Наведемо послідовність кроків для отримання висновку із заданої множини гіпотез:

- 1) $p \rightarrow q$ – гіпотеза;
- 2) $\bar{q} \rightarrow \bar{p}$ – контрапозиція;
- 3) $\bar{p} \rightarrow r$ – гіпотеза;
- 4) $\bar{q} \rightarrow r$ – гіпотетичний силогізм до 2 та 3;
- 5) $r \rightarrow s$ – гіпотеза;
- 6) $\bar{q} \rightarrow s$ – гіпотетичний силогізм до 4 та 5. Висновок доведено [11].

3.3. Алгоритмізація за темою роботи

На головній сторінці тренажеру користувачеві Виводиться назва тренажеру, інформація про розробника і керівника. Також надається можливість ознайомитися з теоретичним матеріалом, в якому наведено і приклади, та перейти до проходження [2].

Тренажер містить два типи завдань: на встановлення відповідності і вибір відповіді з наведених варіантів. Тому розглянемо алгоритм кожного з таких завдань.

Правила виведення в численні висловлювань.

Спочатку відображається умова: «Правила виведення обґрунтовують кроки доведення логічних теорем, яке полягає в перевірці того, що висновок являє собою логічний наслідок множини гіпотез».

Крок 1. Користувачеві відображається завдання: «Встановити відповідність між правилами виведення та відповідними їм тавтологіями». Виводяться правила виведення та тавтології:

Правило виведення	Тавтологія
$p \mid -p \vee q$	$(p \wedge (p \rightarrow q)) \rightarrow q$
$p, q \mid -p \wedge q$	$p \rightarrow (p \vee q)$
$p, p \rightarrow q \mid -q$	$(\bar{q} \wedge (p \rightarrow q)) \rightarrow \bar{p}$
$\bar{q}, p \rightarrow q \mid -\bar{p}$	$((p) \wedge (q)) \rightarrow (p \wedge q)$

Якщо користувач вірно встановлює відповідність, то відбувається перехід на наступний крок, якщо ні – виводиться повідомлення про помилку: «Помилка! Правила виведення та відповідні їм тавтології:

Правило виведення	Тавтологія
$p \mid -p \vee q$	$p \rightarrow (p \vee q)$
$p, q \mid -p \wedge q$	$((p) \wedge (q)) \rightarrow (p \wedge q)$
$p, p \rightarrow q \mid -q$	$(p \wedge (p \rightarrow q)) \rightarrow q$
$\bar{q}, p \rightarrow q \mid -\bar{p}$	$(\bar{q} \wedge (p \rightarrow q)) \rightarrow \bar{p}$

».

Крок 2. Користувачеві відображається завдання: «Встановити відповідність між правилами виведення та відповідними їм тавтологіями». Виводяться правила виведення та тавтології:

Правило виведення	Тавтологія
$p \wedge q -p$	$((p \vee q) \wedge (\bar{p} \vee r)) \rightarrow (q \vee r)$
$p \rightarrow q, q \rightarrow r -p \rightarrow r$	$((p \vee q) \wedge \bar{p}) \rightarrow q$
$p \vee q, \bar{p} -q$	$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$
$p \vee q, \bar{p} \vee r -q \vee r$	$(p \wedge q) \rightarrow p$

Якщо користувач вірно встановлює відповідність, то відбувається перехід на наступний крок, якщо ні – виводиться повідомлення про помилку: «Помилка! Правила виведення та відповідні їм тавтології:

Правило виведення	Тавтологія
$p \wedge q -p$	$(p \wedge q) \rightarrow p$
$p \rightarrow q, q \rightarrow r -p \rightarrow r$	$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$
$p \vee q, \bar{p} -q$	$((p \vee q) \wedge \bar{p}) \rightarrow q$
$p \vee q, \bar{p} \vee r -q \vee r$	$((p \vee q) \wedge (\bar{p} \vee r)) \rightarrow (q \vee r)$

».

Крок 3. Користувачеві відображається завдання: «Встановити відповідність між правилами виведення та їх назвами». Виводяться правила виведення та тавтології:

Правило виведення	Назва правила
$p -p \vee q$	Виключення кон'юнкції
$p \wedge q -p$	Уведення диз'юнкції
$p, q -p \wedge q$	Modus ponens
$p, p \rightarrow q -q$	Уведення кон'юнкції

Якщо користувач вірно встановлює відповідність, то відбувається перехід на наступний крок, якщо ні – виводиться повідомлення про помилку: «Помилка! Правила виведення та їх назви:

Правило виведення	Назва правила
$p \mid -p \vee q$	Уведення диз'юнкції
$p \wedge q \mid -p$	Виключення кон'юнкції
$p, q \mid -p \wedge q$	Уведення кон'юнкції
$p, p \rightarrow q \mid -q$	Modus ponens

».

Крок 4. Користувачеві відображається завдання: «Встановити відповідність між правилами виведення та їх назвами». Виводяться правила виведення та тавтології:

Правило виведення	Назва правила
$\bar{q}, p \rightarrow q \mid -\bar{p}$	Гіпотетичний силогізм
$p \rightarrow q, q \rightarrow r \mid -p \rightarrow r$	Диз'юнктивний силогізм
$p \vee q, \bar{p} \mid -q$	Резолюція
$p \vee q, \bar{p} \vee r \mid -q \vee r$	Modus tollens

Якщо користувач вірно встановлює відповідність, то відбувається перехід на наступний крок, якщо ні – виводиться повідомлення про помилку: «Помилка! Правила виведення та їх назви:

Правило виведення	Назва правила
$\bar{q}, p \rightarrow q \mid -\bar{p}$	Modus tollens
$p \rightarrow q, q \rightarrow r \mid -p \rightarrow r$	Гіпотетичний силогізм
$p \vee q, \bar{p} \mid -q$	Диз'юнктивний силогізм
$p \vee q, \bar{p} \vee r \mid -q \vee r$	Резолюція

».

Крок 5. Користувачеві виводиться повідомлення: «Вітаємо! Ви ознайомилися з темою «Правила виведення в численні висловлювань». Спробуйте застосувати отриманні знання на прикладах.» та надається

можливість перейти до розв'язування прикладів. Також можливо завершити роботу тренажеру.

Застосування правил виведення в численні висловлювань.

Крок 1. Користувачеві відображається завдання: «Яке правило виведення використано в міркуванні «Похолоднішало. Отже, похолоднішало чи почав падати дощ»? Нехай p – це висловлювання «Похолоднішало», а q – висловлювання «Почав падати дощ». Виберіть правило виведення». Наводяться варіанти відповіді:

- Твердження можна записати у вигляді правила введення диз'юнкції $p \mid -p \vee q$;
- Твердження можна записати у вигляді правила виключення кон'юнкції $p \wedge q \mid -p$;
- Твердження можна записати у вигляді правила введення кон'юнкції $p, q \mid -p \wedge q$;
- Твердження можна записати у вигляді правила диз'юнктивного силізму $p \vee q, \bar{p} \mid -q$.

Якщо користувач обирає вірну відповідь, перший варіант, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Твердження можна записати у вигляді правила введення диз'юнкції $p \mid -p \vee q$ ».

Крок 2. Користувачеві відображається завдання: «Яке правило виведення використано в міркуванні «Похолоднішало та почав падати дощ. Отже, похолоднішало»? Нехай p – це висловлювання «Похолоднішало», а q – висловлювання «Почав падати дощ». Виберіть правило виведення». Наводяться варіанти відповіді:

- Твердження можна записати у вигляді правила введення диз'юнкції $p \mid -p \vee q$;

- Твердження можна записати у вигляді правила виключення кон'юнкції $p \wedge q | -p$;
- Твердження можна записати у вигляді правила введення кон'юнкції $p, q | -p \wedge q$;
- Твердження можна записати у вигляді правила диз'юнктивного силогізму $p \vee q, \bar{p} | -q$.

Якщо користувач обирає вірну відповідь, другий варіант, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Твердження можна записати у вигляді правила виключення кон'юнкції $p \wedge q | -p$ ».

Крок 3. Користувачеві відображається завдання: «Яке правило виведення використано в міркуванні «Якщо сьогодні буде дощ, то сьогодні ми не поїдемо на пікнік. Якщо ми не поїдемо на пікнік сьогодні, то поїдемо на пікнік завтра. Отже, якщо сьогодні буде дощ, то ми поїдемо на пікнік завтра.»?

Нехай p : «Сьогодні буде дощ», q : «Сьогодні ми не поїдемо на пікнік», а r : «Ми поїдемо на пікнік завтра». Виберіть правило виведення». Наводяться варіанти відповіді:

- Твердження можна записати у вигляді правила *modus ponens* $p, p \rightarrow q | -q$;
- Твердження можна записати у вигляді правила $p \rightarrow q, q \rightarrow r | -p \rightarrow r$ гіпотетичного силогізму;
- Твердження можна записати у вигляді правила резолюції $p \vee q, \bar{p} \vee r | -q \vee r$;
- Твердження можна записати у вигляді правила $p \vee q, \bar{p} | -q$ диз'юнктивного силогізму.

Якщо користувач обирає вірну відповідь, другий варіант, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Твердження можна записати у вигляді правила $p \rightarrow q, q \rightarrow r \mid -p \rightarrow r$ гіпотетичного силлогізму».

Крок 4. Користувачеві відображається умова: «Довести, що з гіпотез «Сьогодні не сонячний день і холодніше, ніж учора», «Ми підемо купатися лише якщо сьогодні сонячний день», «Якщо ми не підемо купатись, то поїдемо плавати на човні» та «Якщо ми поїдемо плавати на човні, то повернемося пізно ввечері» випливає висновок «Ми повернемося пізно ввечері».

Нехай p : «Сьогодні сонячний день», q : «Сьогодні холодніше, ніж учора», r : «Ми підемо купатися», s : «Ми поїдемо плавати на човні», t : «Ми повернемося пізно ввечері».

Виводиться завдання: «Виберіть формулу гіпотези «Сьогодні не сонячний день і холодніше, ніж учора». Наводяться варіанти відповіді:

- $p \wedge q$;
- $\bar{p} \vee q$;
- $p \vee q$;
- $\bar{p} \wedge q$.

Якщо користувач обирає вірну відповідь, четвертий варіант, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Гіпотеза «Сьогодні не сонячний день і холодніше, ніж учора» – $\bar{p} \wedge q$ ».

Крок 5. Користувачеві відображається завдання: «Виберіть формулу гіпотези «Ми підемо купатися лише якщо сьогодні сонячний день». Наводяться варіанти відповіді:

- $r \wedge p$;
- $r \rightarrow \bar{p}$;

- $r \rightarrow p$;
- $\bar{r} \rightarrow p$.

Якщо користувач обирає вірну відповідь, третій варіант, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Гіпотеза «Ми підемо купатися лише якщо сьогодні сонячний день» – $r \rightarrow p$ ».

Крок 6. Користувачеві відображається завдання: «Виберіть формулу гіпотези «Якщо ми не підемо купатись, то поїдемо плавати на човні». Наводяться варіанти відповіді:

- $\bar{r} \rightarrow s$;
- $\bar{r} \rightarrow \bar{s}$;
- $r \rightarrow s$;
- $r \rightarrow \bar{s}$.

Якщо користувач обирає вірну відповідь, перший варіант, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Гіпотеза «Якщо ми не підемо купатись, то поїдемо плавати на човні» – $\bar{r} \rightarrow s$ ».

Крок 7. Користувачеві відображається завдання: «Виберіть формулу гіпотези «Якщо ми поїдемо плавати на човні, то повернемося пізно ввечері». Наводяться варіанти відповіді:

- $s \rightarrow t$;
- $\bar{s} \rightarrow t$;
- $s \rightarrow \bar{t}$;
- $\bar{s} \rightarrow \bar{t}$.

Якщо користувач обирає вірну відповідь, перший варіант, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Гіпотеза «Якщо ми поїдемо плавати на човні, то повернемося пізно ввечері» – $s \rightarrow t$ ».

Крок 8. Користувачеві відображається питання: «Чи доведено висновок?». Наводяться варіанти відповіді:

- Так, висновок доведено;
- Ні, висновок не доведено.

Якщо користувач обирає вірну відповідь, другий варіант, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Висновок не доведено».

Крок 9. Користувачеві відображається завдання: «Гіпотези можна записати у вигляді $\bar{p} \wedge q$, $r \rightarrow p$, $\bar{r} \rightarrow s$, $s \rightarrow t$, а висновок – t . Встановити послідовність отримання висновку із заданої множини гіпотез із зазначенням застосованих правил виведення». Наводиться перелік кроків:

- $\bar{p} \wedge q$ – гіпотеза;
- s – modus ponens до 4 та 5;
- $\bar{r} \rightarrow s$ – гіпотеза;
- $r \rightarrow p$ – гіпотеза;
- t – modus ponens до 6 та 7;
- \bar{p} – правило виключення кон'юнкції до 1;
- $s \rightarrow t$ – гіпотеза;
- \bar{r} – modus tollens до 2 та 3.

Якщо користувач встановлює вірну послідовність, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Послідовність отримання висновку із заданої множини гіпотез із зазначенням застосованих правил виведення має наступний вигляд:

- 1) $\bar{p} \wedge q$ – гіпотеза;

- 2) \bar{p} – правило виключення кон'юнкції до 1;
- 3) $r \rightarrow p$ – гіпотеза;
- 4) \bar{r} – modus tollens до 2 та 3;
- 5) $\bar{r} \rightarrow s$ – гіпотеза;
- 6) s – modus ponens до 4 та 5;
- 7) $s \rightarrow t$ – гіпотеза;
- 8) t – modus ponens до 6 та 7. ».

Крок 10. Користувачеві відображається питання: «Чи доведено висновок?». Наводяться варіанти відповіді:

- Так, висновок доведено;
- Ні, висновок не доведено.

Якщо користувач обирає вірну відповідь, перший варіант, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Висновок доведено».

Крок 11. Користувачеві відображається умова: «Довести, що з гіпотез «Якщо ти надішлеш мені повідомлення електронною поштою, то я закінчу писати програму», «Якщо ти не надішлеш мені повідомлення електронною поштою, то я рано піду спати» та «Якщо я рано піду спати, то прокинуся бадьорим» впливає висновок «Якщо я не закінчу писати програму, то я прокинуся бадьорим».

Нехай p : «Ти надішлеш мені повідомлення електронною поштою», q : «Я закінчу писати програму», r : «Я рано піду спати», s : «Я прокинуся бадьорим».

Виводиться завдання: «Виберіть формулу гіпотези «Якщо ти надішлеш мені повідомлення електронною поштою, то я закінчу писати програму». Наводяться варіанти відповіді:

- $p \rightarrow q$;
- $\bar{p} \rightarrow q$;

- $p \rightarrow \bar{q}$;
- $p \wedge q$.

Якщо користувач обирає вірну відповідь, перший варіант, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Гіпотеза «Якщо ти надішлеш мені повідомлення електронною поштою, то я закінчу писати програму» – $p \rightarrow q$ ».

Крок 12. Користувачеві відображається завдання: «Виберіть формулу гіпотези «Якщо ти не надішлеш мені повідомлення електронною поштою, то я рано піду спати». Наводяться варіанти відповіді:

- $\bar{p} \rightarrow r$;
- $p \rightarrow r$;
- $\bar{p} \rightarrow \bar{r}$;
- $p \rightarrow \bar{r}$.

Якщо користувач обирає вірну відповідь, перший варіант, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Гіпотеза «Якщо ти не надішлеш мені повідомлення електронною поштою, то я рано піду спати» – $\bar{p} \rightarrow r$ ».

Крок 13. Користувачеві відображається завдання: «Виберіть формулу гіпотези «Якщо я рано піду спати, то прокинуся бадьорим». Наводяться варіанти відповіді:

- $\bar{r} \rightarrow s$;
- $r \rightarrow \bar{s}$;
- $r \rightarrow s$;
- $r \vee s$.

Якщо користувач обирає вірну відповідь, третій варіант, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Гіпотеза «Якщо я рано піду спати, то прокинуся бадьорим» – $r \rightarrow s$ ».

Крок 14. Користувачеві відображається завдання: «Виберіть формулу висновку «Якщо я не закінчу писати програму, то я прокинуся бадьорим». Наводяться варіанти відповіді:

- $q \rightarrow \bar{s}$;
- $q \rightarrow s$;
- $\bar{q} \rightarrow s$;
- $\bar{q} \rightarrow \bar{s}$.

Якщо користувач обирає вірну відповідь, третій варіант, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Висновок «Якщо я не закінчу писати програму, то я прокинуся бадьорим» – $\bar{q} \rightarrow s$ ».

Крок 15. Користувачеві відображається питання: «Чи доведено висновок?». Наводяться варіанти відповіді:

- Так, висновок доведено;
- Ні, висновок не доведено.

Якщо користувач обирає вірну відповідь, другий варіант, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Висновок не доведено».

Крок 16. Користувачеві відображається завдання: «Гіпотези можна записати у вигляді, $p \rightarrow q$, $\bar{p} \rightarrow r$, $r \rightarrow s$. Потрібно обґрунтувати висновок $\bar{q} \rightarrow s$. Встановити послідовність отримання висновку із заданої множини гіпотез». Наводиться перелік кроків:

- $\bar{p} \rightarrow r$ – гіпотеза;
- $p \rightarrow q$ – гіпотеза;

- $\bar{q} \rightarrow \bar{p}$ – контрапозиція;
- $r \rightarrow s$ – гіпотеза;
- $\bar{q} \rightarrow s$ – гіпотетичний силогізм до 4 та 5;
- $\bar{q} \rightarrow r$ – гіпотетичний силогізм до 2 та 3.

Якщо користувач встановлює вірну послідовність, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Послідовність отримання висновку із заданої множини має наступний вигляд:

- 1) $p \rightarrow q$ – гіпотеза;
- 2) $\bar{q} \rightarrow \bar{p}$ – контрапозиція;
- 3) $\bar{p} \rightarrow r$ – гіпотеза;
- 4) $\bar{q} \rightarrow r$ – гіпотетичний силогізм до 2 та 3;
- 5) $r \rightarrow s$ – гіпотеза;
- 6) $\bar{q} \rightarrow s$ – гіпотетичний силогізм до 4 та 5. ».

Крок 17. Користувачеві відображається питання: «Чи доведено висновок?».

Наводяться варіанти відповіді:

- Так, висновок доведено;
- Ні, висновок не доведено.

Якщо користувач обирає вірну відповідь, перший варіант, то відбувається перехід на наступний крок.

Якщо ні – виводиться повідомлення про помилку: «Помилка! Висновок доведено».

Крок 18. Користувачеві виводиться повідомлення: «Вітаємо! Ви ознайомилися з темою «Застосування правил виведення в численні висловлювань». Пропонується завершити роботу тренажеру або перейти до початку й пройти тренажер знову.

3.4. Розробка блок-схеми

На рисунку 3.1 зображено блок-схему алгоритму роботи тренажеру.

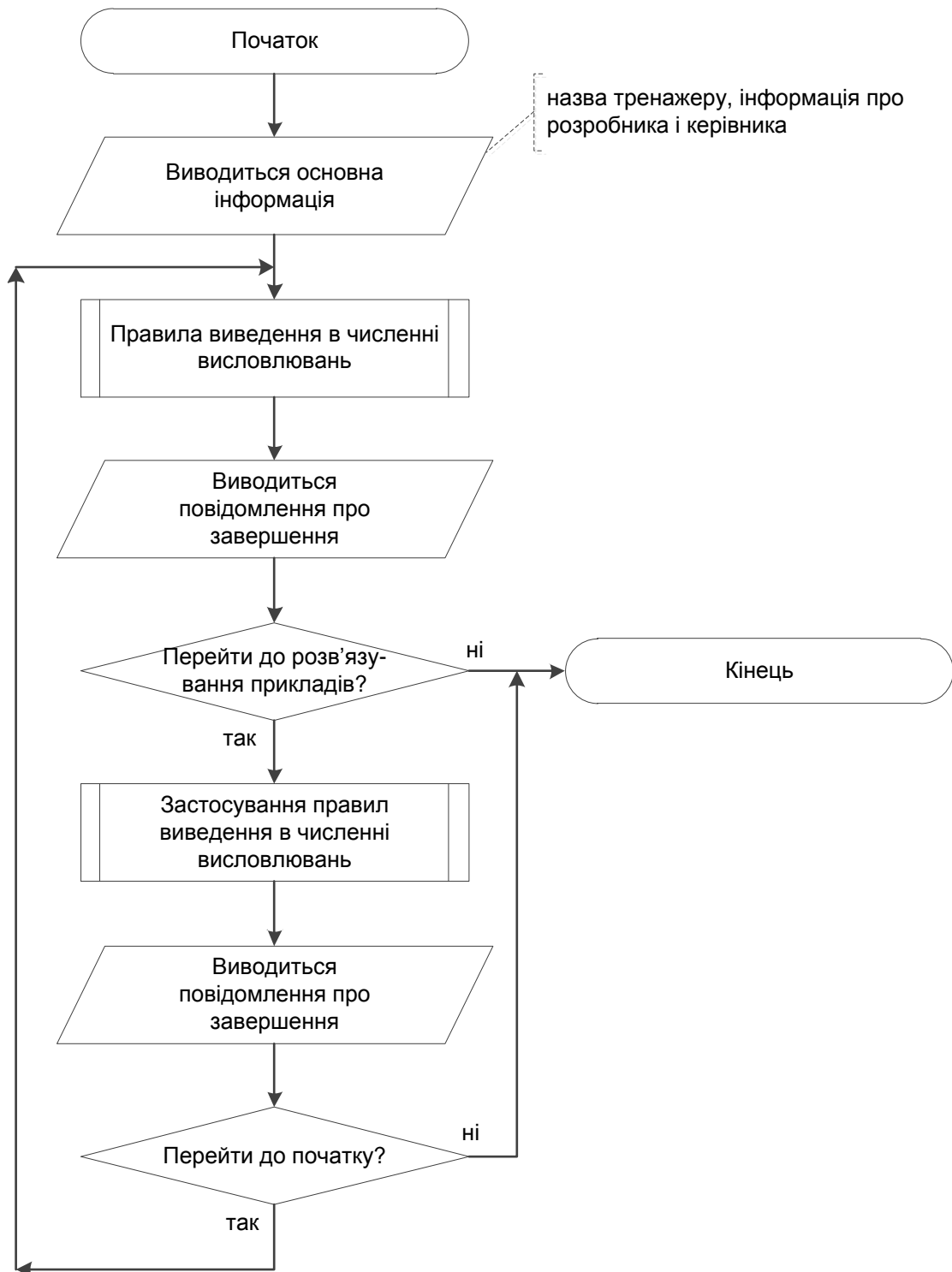


Рисунок 3.1 – Блок-схема алгоритму роботи тренажеру

На рисунку 3.2 зображена блок-схема блоку «Правила виведення в численні висловлювань».

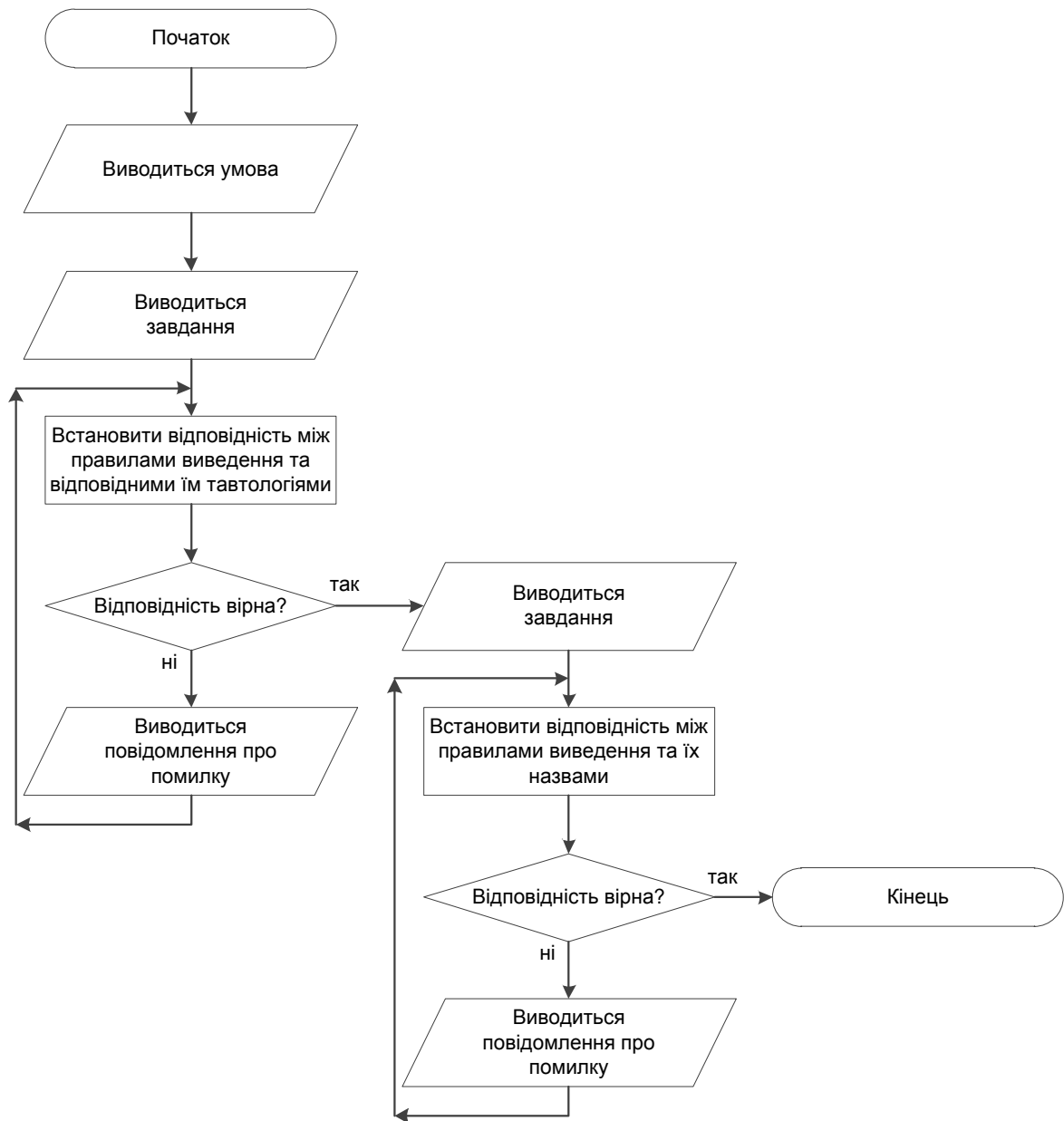


Рисунок 3.2 – Блок-схема блоку «Правила виведення в численні висловлювань»

На рисунках 3.3-3.11 зображена блок-схема блоку «Застосування правил виведення в численні висловлювань».



Рисунок 3.3 – Блок-схема блоку «Застосування правил виведення в численні висловлювань»

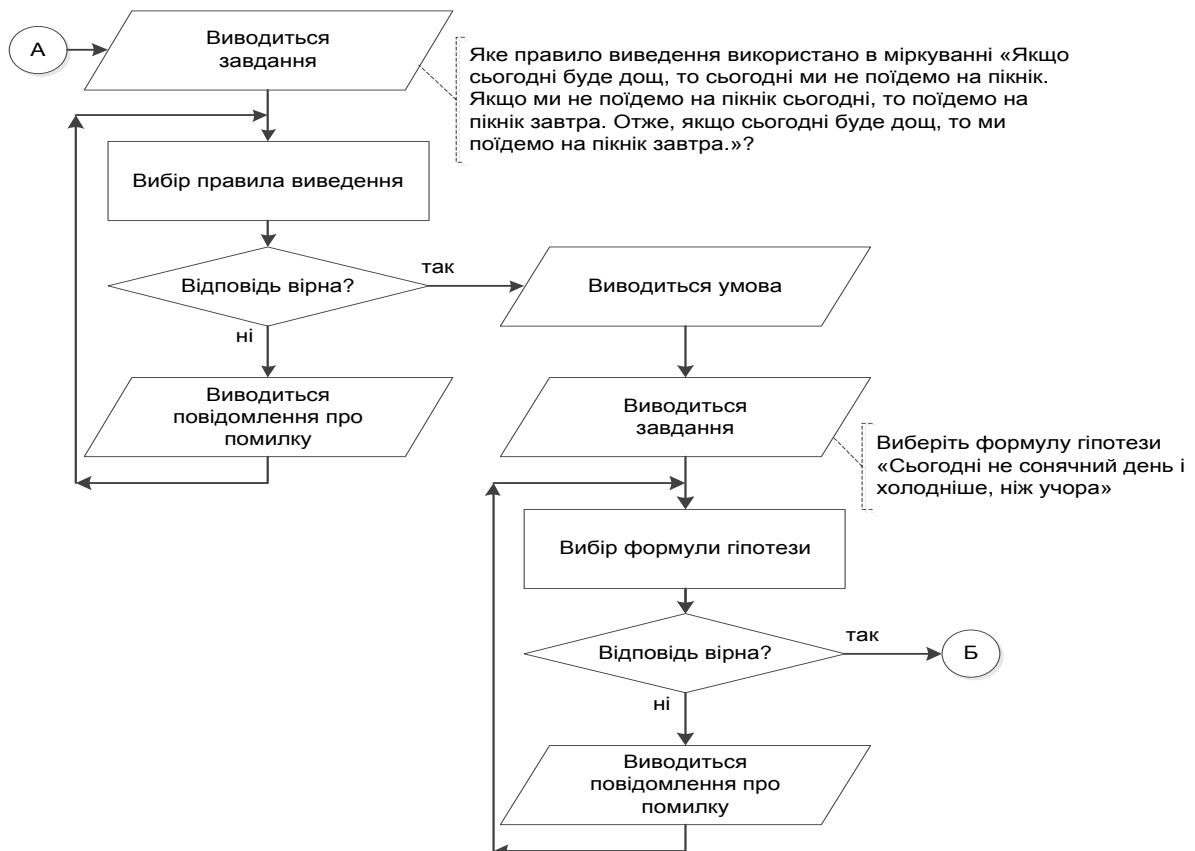


Рисунок 3.4 – Продовження блок-схеми блоку «Застосування правил виведення в численні висловлювань»

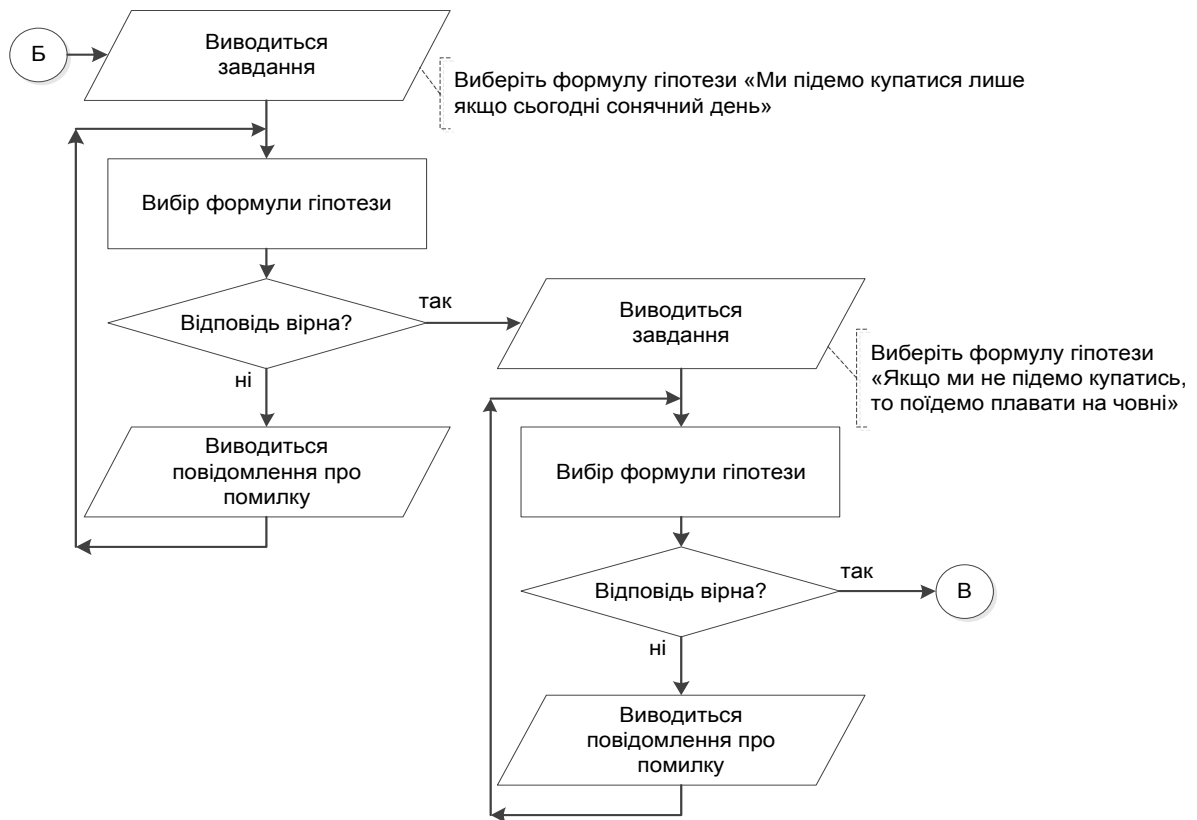


Рисунок 3.5 – Продовження блок-схеми блоку «Застосування правил виведення в численні висловлювань»

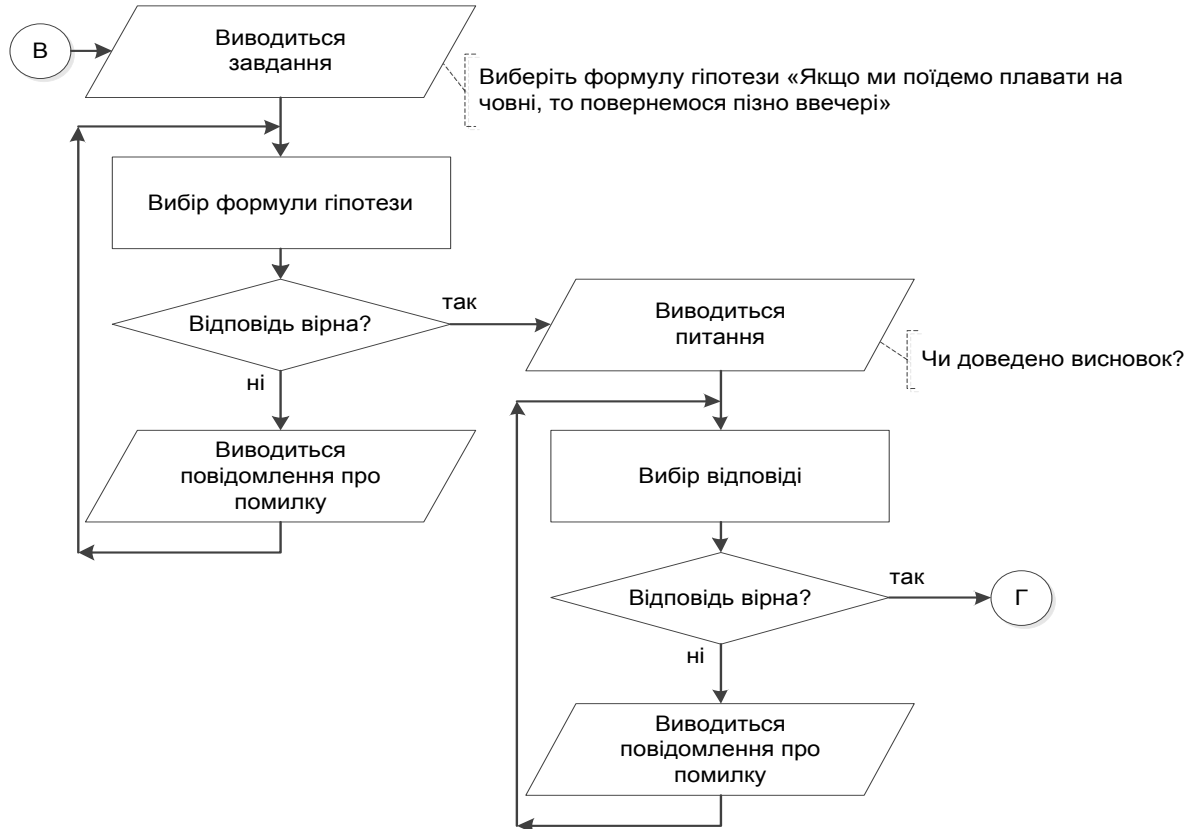


Рисунок 3.6 – Продовження блок-схеми блоку «Застосування правил виведення в численні висловлювань»

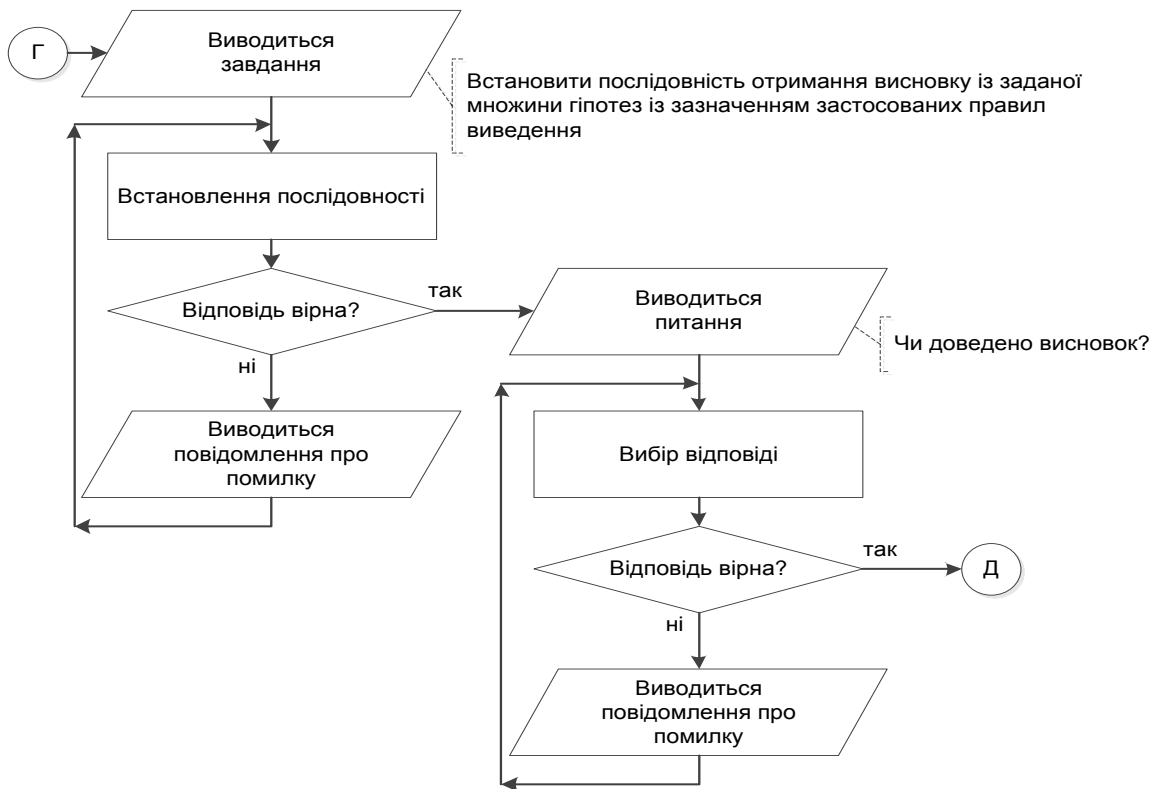


Рисунок 3.7 – Продовження блок-схеми блоку «Застосування правил виведення в численні висловлювань»

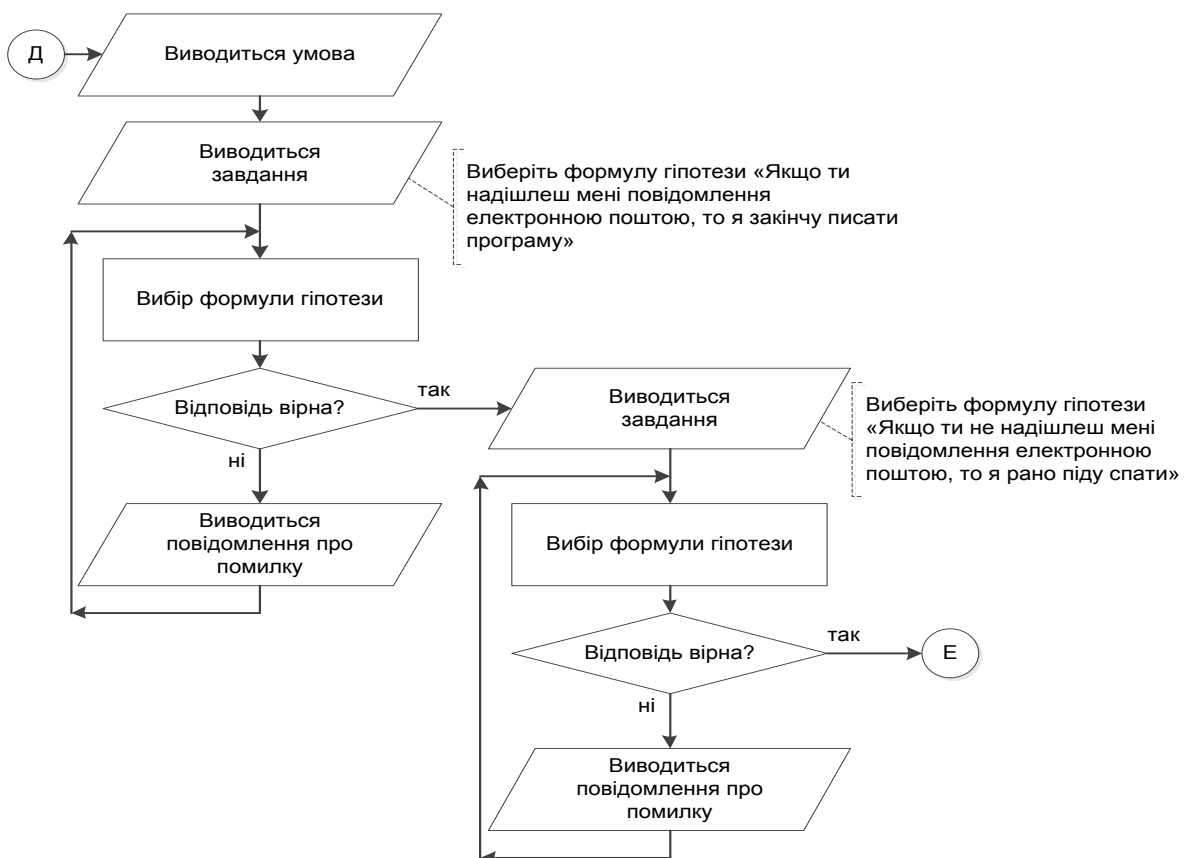


Рисунок 3.8 – Продовження блок-схеми блоку «Застосування правил виведення в численні висловлювань»

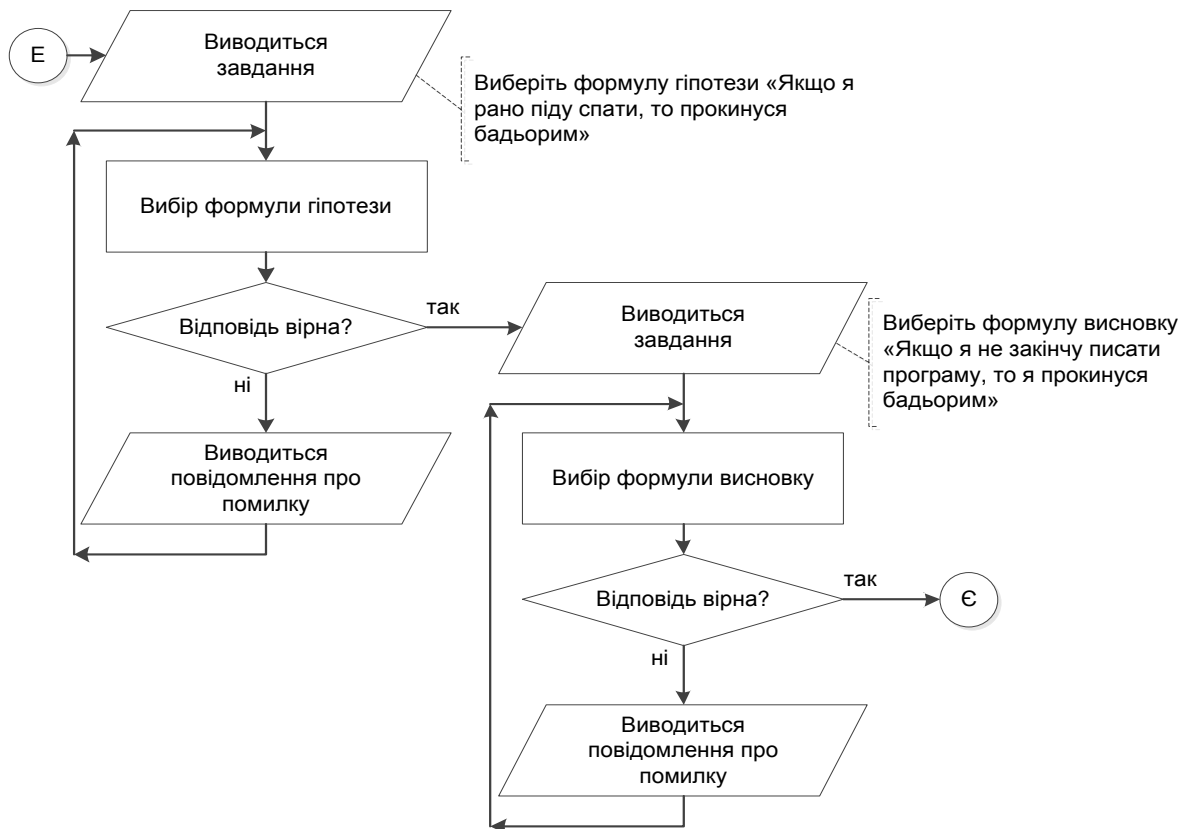


Рисунок 3.9 – Продовження блок-схеми блоку «Застосування правил виведення в численні висловлювань»

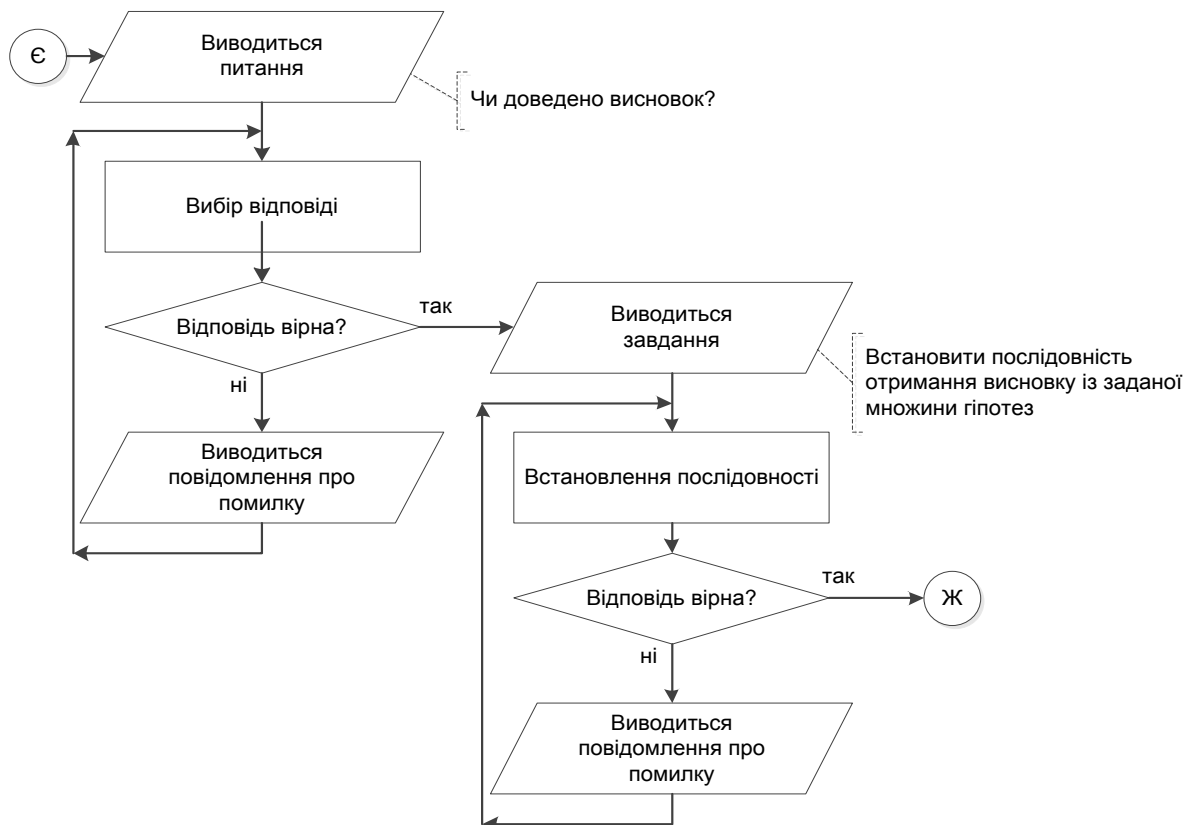


Рисунок 3.10 – Продовження блок-схеми блоку «Застосування правил виведення в численні висловлювань»

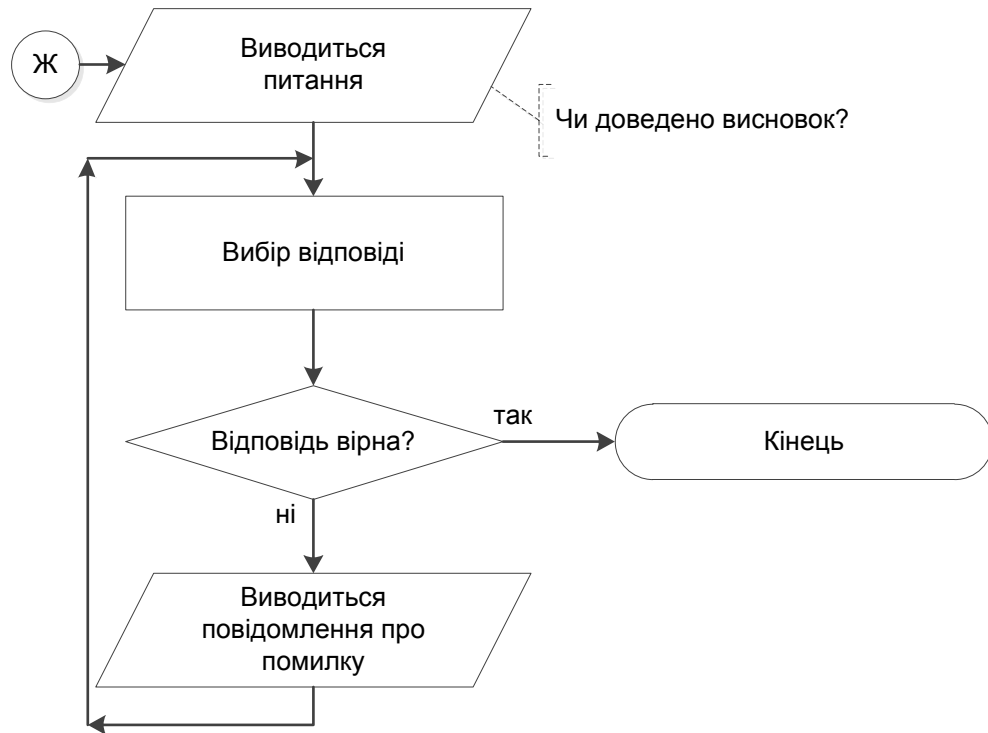


Рисунок 3.11 – Продовження блок-схеми блоку «Застосування правил виведення в численні висловлювань»

4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів

IDE NetBeans дозволяє швидко і легко розробляти настільні, мобільні і веб-додатки Java, а також додатки HTML5 з використанням технологій HTML, JavaScript і CSS. IDE також надає багатофункціональні набори засобів для розробників PHP і C / C ++. Це безкоштовне ПО з відкритим вихідним кодом, яке має велике співтовариство користувачів і розробників по всьому світу.

У порівнянні з іншими середовищами IDE, IDE NetBeans забезпечує висококласну комплексну підтримку новітніх технологій Java і останніх удосконалень стандартів Java. Це перше безкоштовне середовище IDE, що підтримує JDK 8, JDK 7, Java EE 7, включаючи відповідні удосконалення HTML5 і JavaFX 2.

Завдяки редактору Java, різноманітним можливостям і широкого спектру інструментів, шаблонів і зразків, IDE NetBeans встановлює стандарт розробки за допомогою новітніх готових технологій.

IDE - це набагато більше, ніж просто текстовий редактор. У редакторі NetBeans обробляються відступи ліній, зіставляються слова і дужки, а також виділяється вихідний код як синтаксично, так і семантично. Він також надає шаблони коду, поради з написання коду і інструменти реорганізації.

Редактор підтримує безліч мов - від Java, C / C ++, XML і HTML до PHP, Groovy, Javadoc, JavaScript і JSP. Редактор є розширюваним, тому можна підключити підтримку для безлічі інших мов.

IDE NetBeans пропонує різні уявлення даних, від декількох вікон проектів до корисних засобів настройки та ефективного управління додатками, що забезпечує швидкий і зручний перехід за рівнями ієрархії

даних, а також надає засоби управління версіями за допомогою готової інтеграції Subversion, Mercurial і Git.

За допомогою редакторів і функцій перетягування в середовищі IDE можна швидко і ефективно розробляти графічні інтерфейси для додатків Java SE, HTML5, Java EE, PHP, C / C ++ і Java ME.

NetBeans GUI Builder автоматично обробляє правильні інтервали між рядками і вирівнювання для додатків Java SE, підтримуючи редагування всередині рядка.

NetBeans надає методи статичного аналізу, зокрема інтеграцію з інструментом FindBugs, для виявлення та виправлення поширених проблем в коді Java. Крім того, відладчик NetBeans дозволяє розміщувати точки зупину в вихідному коді, додавати контрольні модулі для полів, виконувати код в покроковому режимі з заходом в методи, створювати знімки і відстежувати виконання [12].

Java – популярна мова програмування, яка з’явилася в 1995 році й наразі налічує дев’ять основних версій. Java стала технічним феноменом, що багато в чому пов’язано з її унікальною портативністю: програми Java працюють на будь-якому пристрої або операційній системі.

Java не була першою для написання кросплатформних програм, але стала найпопулярнішою: ідея «написав раз – запускай скрізь» дозволила залучити багато нових облич до розвитку мови. Java-програмісти можуть розробляти застосунок на комп’ютері, а потім відкривати його на цільовій платформі – телефоні, сервері тощо. Якщо для компілятора доступні потрібні бібліотеки, код буде працювати.

Бібліотеки з відкритим кодом полегшують використання Java у всьому світі. Apache, Google та інші організації розробили велику кількість потужних бібліотек, що полегшує та пришвидшує розробку програм [13-14].

4.2. Опис процесу програмної реалізації

Так як тренажер реалізовано у вигляді Java-аплету, то ініціалізація програми і відображення аплета відбувається за допомогою наступного коду:

```
@Override
public void init() {
    /* Create and display the applet */
    try {
        java.awt.EventQueue.invokeLaterAndWait(new Runnable() {
            public void run() {
                initComponents();
            }
        });
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
```

Наступним кроком створено змінну `int` і типу `Integer` та `CardLayout` `card`.

`CardLayout card` – використовується для переключення між панелями програми.

`int i=1` – вказує поточний крок алгоритму, присвоєно значення 1.

Потім розроблено графічне представлення тренажеру. Так панель `Main [JTabbedPane]` має три вкладки:

- `Start [JPanel]` – стартова сторінка;
- `Theory [JTabbedPane]` – теоретичний матеріал, поділений на вкладки:
 - `TheoryText [JPanel]` – формальна теорія;
 - `TheoryExample1 [JPanel]` – приклад 1;

- TheoryExample2 [JPanel] – приклад 2;
- TheoryExample3 [JPanel] – приклад 3;
- TheoryExample4 [JPanel] – приклад 4;
- Training [JPanel] – тренінг, містить дві панелі:
 - Example1 [JPanel] – кроки блоку «Правила виведення в численні висловлювань»;
 - Example2 [JPanel] – кроки блоку «Застосування правил виведення в численні висловлювань».

Повідомлення про помилку знаходяться на панелях Ex1Error [JDialog], Ex2Error [JDialog], Ex2Error9 [JDialog], Ex2Error16 [JDialog], а про завершення виконання – End [JDialog].

Для відображення наступного кроку алгоритму або повідомлення про виконання було створено функцію `public void example1Next()` для першого блоку питань і `public void example2Next()` для другого. Якщо поточний крок не останній, то виводиться наступне завдання, інакше – повідомлення про завершення.

```
public void example1Next() {
    i++;
    if (i<5) {
        card =(CardLayout) Ex1Steps.getLayout();
        card.show(Ex1Steps, "ex1step"+i);
    } else {
        End.setSize(575, 255);
        End.setLocationRelativeTo(Main);
        card =(CardLayout) Ends.getLayout();
        card.show(Ends, "ex1end");
        End.setVisible(true);
    }
}
```

```

public void example2Next() {
    i++;
    if (i<18) {
        card =(CardLayout) Ex2Steps.getLayout();
        card.show(Ex2Steps, "ex2step"+i);
    } else {
        End.setSize(575, 255);
        End.setLocationRelativeTo(Main);
        card =(CardLayout) Ends.getLayout();
        card.show(Ends, "ex2end");
        End.setVisible(true);
    }
}

```

Аналогічно створено функції `public void example1Error(int step)` і `public void example2Error(int step)` для виведення помилки, лише слід вказати крок алгоритму через параметр `int step` типу `Integer`.

```

public void example1Error(int step) {
    Ex1Error.setSize(465, 255);
    Ex1Error.setLocationRelativeTo(Main);
    card =(CardLayout) Ex1Errors.getLayout();
    card.show(Ex1Errors, "ex1error"+step);
    Ex1Error.setVisible(true);
}

public void example2Error(int step) {
    switch(step) {
        case 9:
            Ex2Error9.setSize(565, 350);
            Ex2Error9.setLocationRelativeTo(Main);
            Ex2Error9.setVisible(true);
            break;

```

```

case 16:
    Ex2Error16.setSize(565, 270);
    Ex2Error16.setLocationRelativeTo(Main);
    Ex2Error16.setVisible(true);
    break;
default:
    Ex2Error.setSize(400, 160);
    Ex2Error.setLocationRelativeTo(Main);
    card =(CardLayout) Ex2Errors.getLayout();
    card.show(Ex2Errors, "ex2error"+step);
    Ex2Error.setVisible(true);
    break;
    }
    }

```

Щоб всі кнопки працювали потрібним чином на кожну з них було написано відповідну подію. Так при натисненні кнопки «Перейти до тренінгу >>>» відбувається подія `private void jButton1ActionPerformed`, що переключає на вкладку з тренінгом. При цьому також блокується вибір мови і якщо вибрано англійську, то відкривається тренажер, в якому весь зміст на англійській мові.

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    Main.setSelectedIndex(2);
    Main.setEnabledAt(2, true);
    Ukr.setEnabled(false);
    Eng.setEnabled(false);
    if(Eng.isSelected()) {
        Output_rules.StartEngApplet();
    }
}

```


Натиснувши кнопку «Продовжити», перевіряється відповідь і якщо вірно, то виконується функція `example1Next()` для блоку «Правила виведення в численні висловлювань» або `example2Next()` для «Застосування правил виведення в численні висловлювань». Якщо ні – `example1Error(i)` або `example2Error(i)`, де *i* – поточний крок. Відповідні події: `private void Ex1NextActionPerformed`, `private void Ex2NextActionPerformed` (Див. Додаток А).

Для переходу від першого блоку питань до другого застосовується подія `private void Ex1toEx2ActionPerformed`, що виконується при натисканні кнопки «Перейти до розв'язування прикладів».

```
private void Ex1toEx2ActionPerformed(java.awt.event.ActionEvent evt)
{
    jSpinner1.setValue(1);
    jSpinner2.setValue(2);
    jSpinner3.setValue(3);
    jSpinner4.setValue(4);
    jSpinner5.setValue(1);
    jSpinner6.setValue(2);
    jSpinner7.setValue(3);
    jSpinner8.setValue(4);
    jSpinner9.setValue(1);
    jSpinner10.setValue(2);
    jSpinner11.setValue(3);
    jSpinner12.setValue(4);
    jSpinner13.setValue(1);
    jSpinner14.setValue(2);
    jSpinner15.setValue(3);
    jSpinner16.setValue(4);
    card =(CardLayout) Training.getLayout();
    card.show(Training, "example2");
}
```

```

        i=0;
        example2Next();
        End.setVisible(false);
    }

```

Навпаки перехід здійснюється за допомогою `private void Ex2toEx1ActionPerformed`, для цього потрібно натиснути кнопку «Перейти до початку й пройти тренажер знову».

```

private void Ex2toEx1ActionPerformed(java.awt.event.ActionEvent evt)
{
    buttonGroup1.clearSelection();
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");
    jTextField6.setText("");
    jTextField7.setText("");
    jTextField8.setText("");
    jTextField9.setText("");
    jTextField10.setText("");
    jTextField11.setText("");
    jTextField12.setText("");
    jTextField13.setText("");
    jTextField14.setText("");
    card =(CardLayout) Training.getLayout();
    card.show(Training, "example1");
    i=0;
    example1Next();
    End.setVisible(false);
}

```

Щоб завершити роботу необхідно натиснути «Завершити роботу тренажеру». При цьому виконається подія `private void Ex1CloseActionPerformed` або `private void Ex2CloseActionPerformed`, так як вони здійснюють одну й ту саму дію, але розміщені на різних панелях.

```
private void Ex1CloseActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}
```

На кожну кнопку зміни актуальної мови написано відповідну подію. Так для «Укр» відбувається подія `private void UkrActionPerformed`, а для Eng – `private void EngActionPerformed` (Див. Додаток А).

4.3. Опис програми

Головна сторінка містить назву тренажеру, інформацію про розробника і керівника (рис. 4.1).

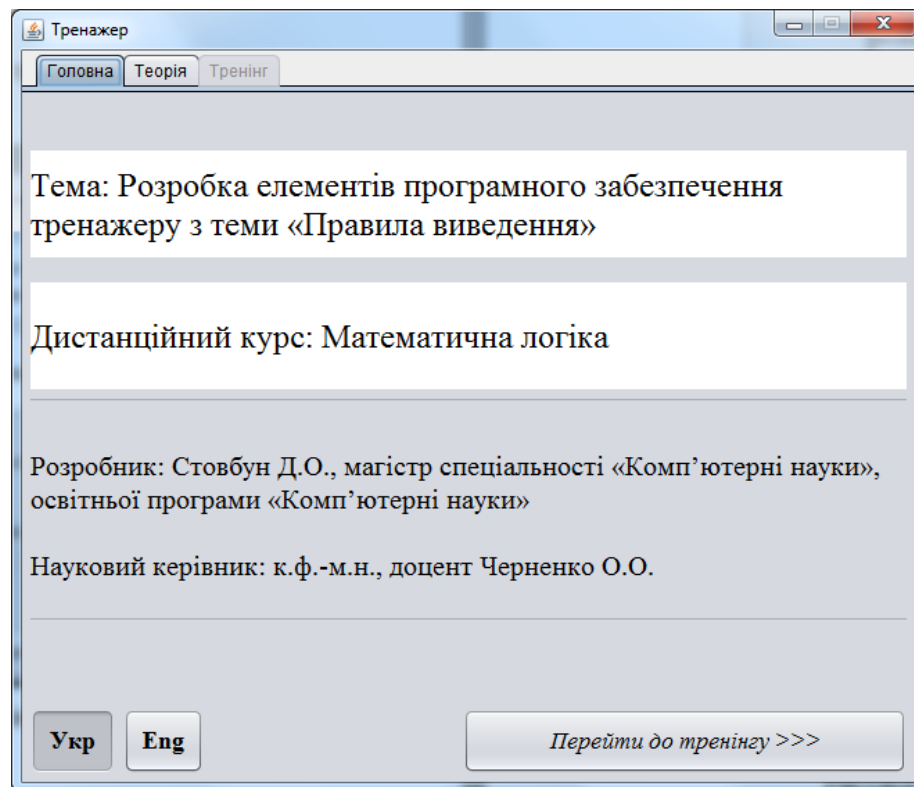


Рисунок 4.1 – Головна сторінка тренажеру

Вкладка «Тренінг» пропонує користувачу виконати спочатку завдання першого блоку, а потім вже другого. У вкладці «Теорія» розміщено теоретичний матеріал, наведено приклади (рис. 4.2, 4.3).

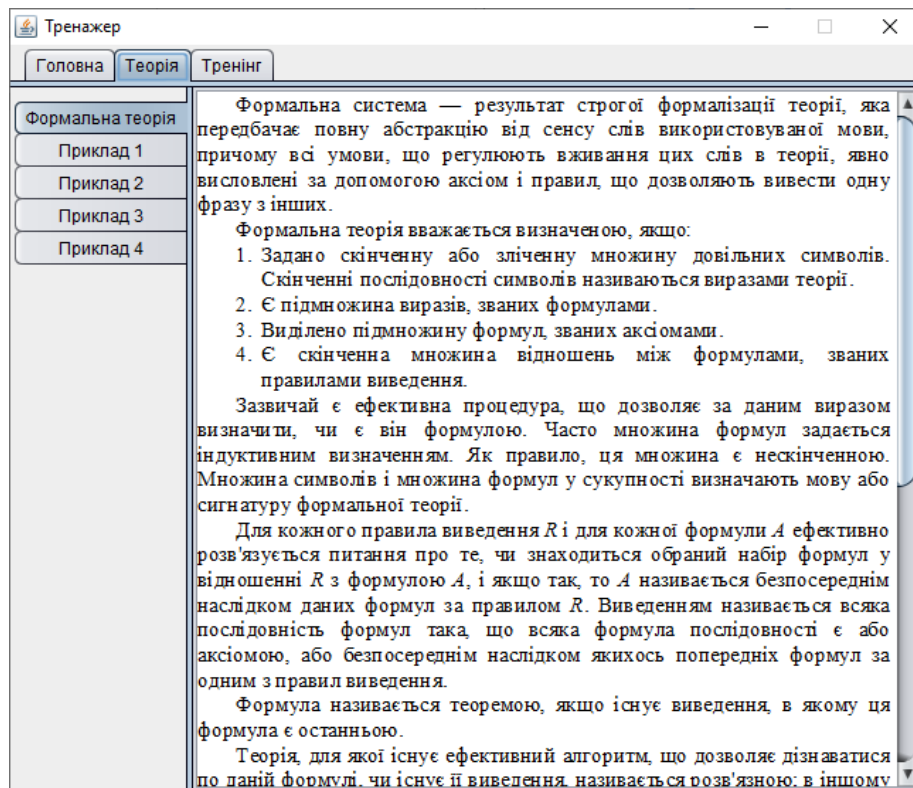


Рисунок 4.2 – Основний матеріал вкладки «Теорія»

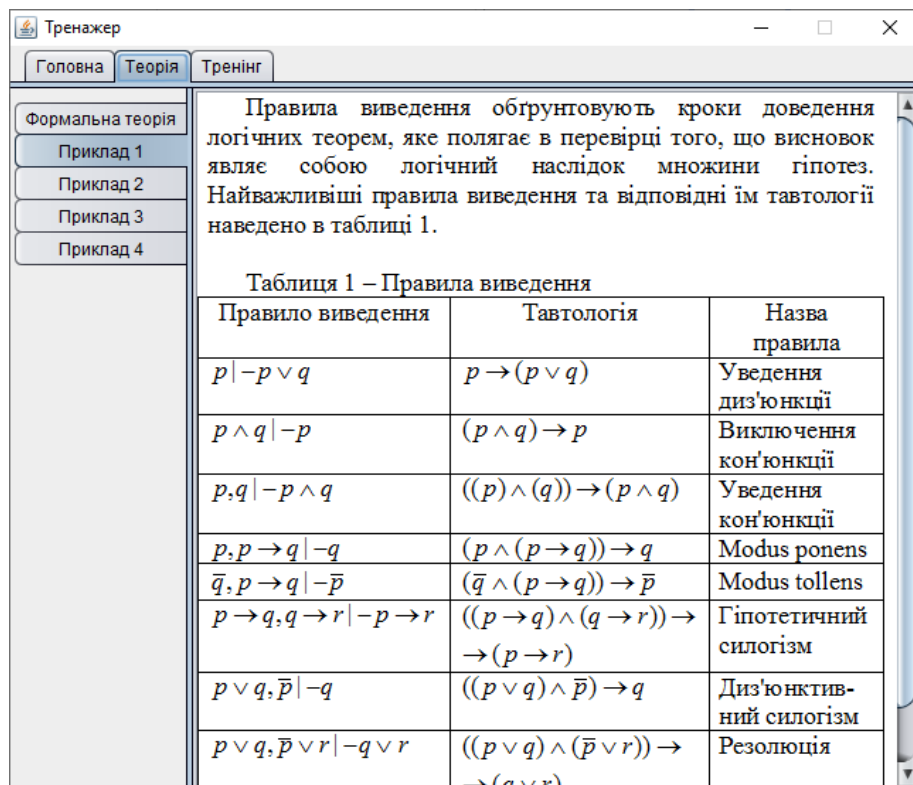


Рисунок 4.3 – Приклад вкладки «Теорія»

Серед всіх завдань програми є декілька видів. Перший – це встановлення відповідності (рис. 4.4).

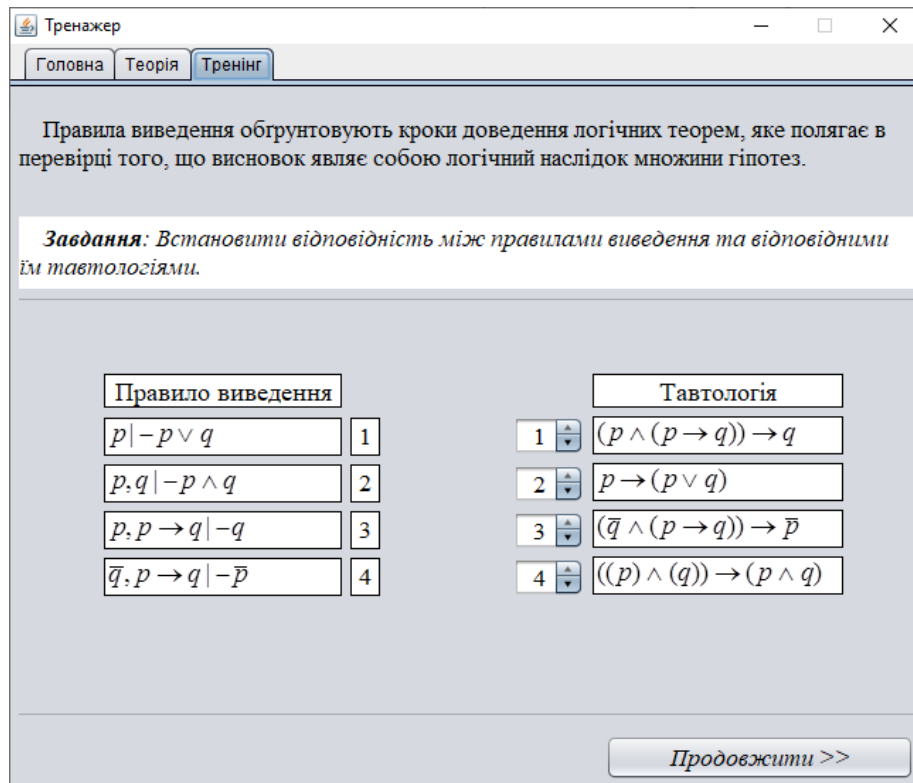


Рисунок 4.4 – Встановлення відповідності

До другого виду відносяться завдання з вибором одного з варіантів відповідей (рис. 4.5).

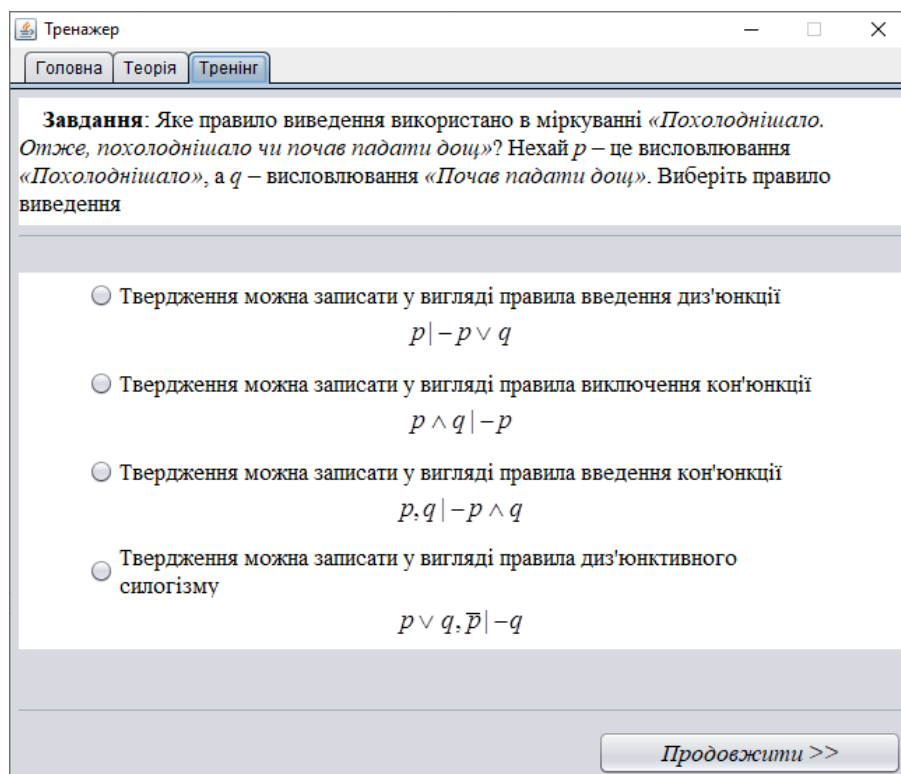


Рисунок 4.5 – Вибір одного з варіантів відповідей

Останній вид – це встановлення послідовності (рис. 4.6).

Тренажер

Головна Теорія Тренінг

Довести, що з гіпотез «Сьогодні не сонячний день і холодніше, ніж учора», «Ми підемо купатися лише якщо сьогодні сонячний день», «Якщо ми не підемо купатись, то поїдемо плавати на човні» та «Якщо ми поїдемо плавати на човні, то повернемося пізно ввечері» випливає висновок «Ми повернемося пізно ввечері».

Нехай p : «Сьогодні сонячний день», q : «Сьогодні холодніше, ніж учора», r : «Ми підемо купатися», s : «Ми поїдемо плавати на човні», t : «Ми повернемося пізно ввечері».

Завдання: Гіпотези можна записати у вигляді

$\bar{p} \wedge q, r \rightarrow p, \bar{r} \rightarrow s, s \rightarrow t$, а висновок – t .

Встановити послідовність отримання висновку із заданої множини гіпотез із зазначенням застосованих правил виведення.

<input type="checkbox"/> $\bar{p} \wedge q$ – гіпотеза	<input type="checkbox"/> t – modus ponens до 6 та 7
<input type="checkbox"/> s – modus ponens до 4 та 5	<input type="checkbox"/> \bar{p} – правило виключення кон'юнкції до 1
<input type="checkbox"/> $\bar{r} \rightarrow s$ – гіпотеза	<input type="checkbox"/> $s \rightarrow t$ – гіпотеза
<input type="checkbox"/> $r \rightarrow p$ – гіпотеза	<input type="checkbox"/> \bar{r} – modus tollens до 2 та 3

Продовжити >>

Рисунок 4.6 – Встановлення послідовності

Якщо відповісти неправильно на будь-яке завдання, то виведеться повідомлення про помилку з вірною відповіддю (рис. 4.7).

Помилка

Помилка! Послідовність отримання висновку із заданої множини гіпотез із зазначенням застосованих правил виведення має наступний вигляд:

- 1) $\bar{p} \wedge q$ – гіпотеза;
- 2) \bar{p} – правило виключення кон'юнкції до 1;
- 3) $r \rightarrow p$ – гіпотеза;
- 4) \bar{r} – modus tollens до 2 та 3;
- 5) $\bar{r} \rightarrow s$ – гіпотеза;
- 6) s – modus ponens до 4 та 5;
- 7) $s \rightarrow t$ – гіпотеза;
- 8) t – modus ponens до 6 та 7.

Рисунок 4.7 – Повідомлення про помилку

При завершенні виконання блоку питань відображається відповідне повідомлення, а також надається можливість перейти до наступного блоку / повернутися на початок, завершити роботу тренажеру (рис. 4.8).

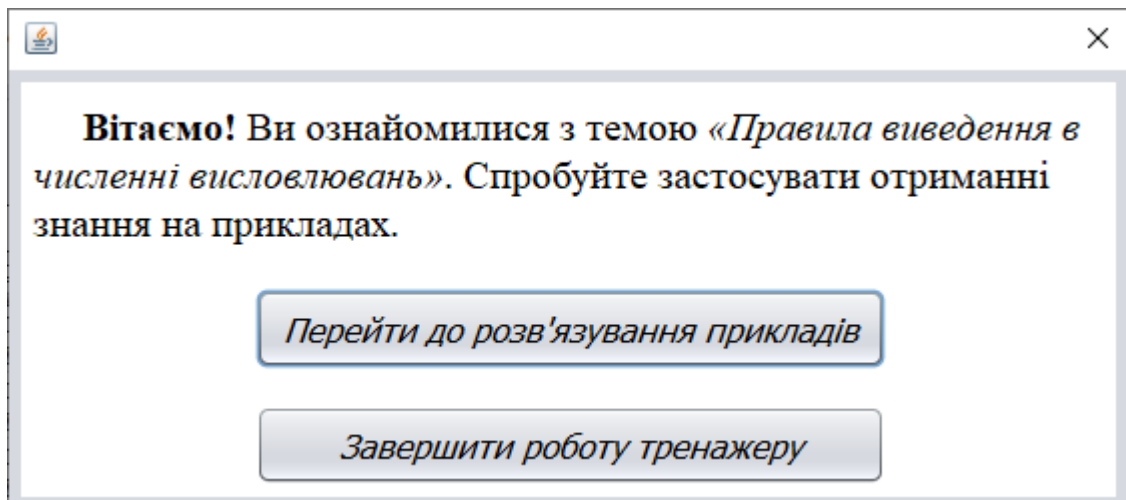


Рисунок 4.8 – Повідомлення про завершення

4.4. Інструкція по використанню тренажеру

Після запуску тренажеру відображається головне вікно (рис.4.1). Щоб ознайомитися з теоретичним матеріалом необхідно натиснути на вкладку «Теорія» (рис. 4.2, 4.3). До даної інформації можна звернутися в будь-який момент.

Щоб змінити мову слід вибрати «Укр» або «Eng», після чого головна сторінка та весь теоретичний матеріал зміниться на вибрану мову (рис. 4.9, 4.10).

Для початку проходження тренажеру потрібно натиснути кнопку «Перейти до тренінгу >>>».

При переході до тренінгу українською мовою відкриється вкладка «Тренінг». Якщо вибрано англійську мову, то запуситься нове вікно тренажеру, де весь вміст вже буде англійською мовою. При цьому закриється попереднє вікно, а також відразу відкриється вкладка «Training».

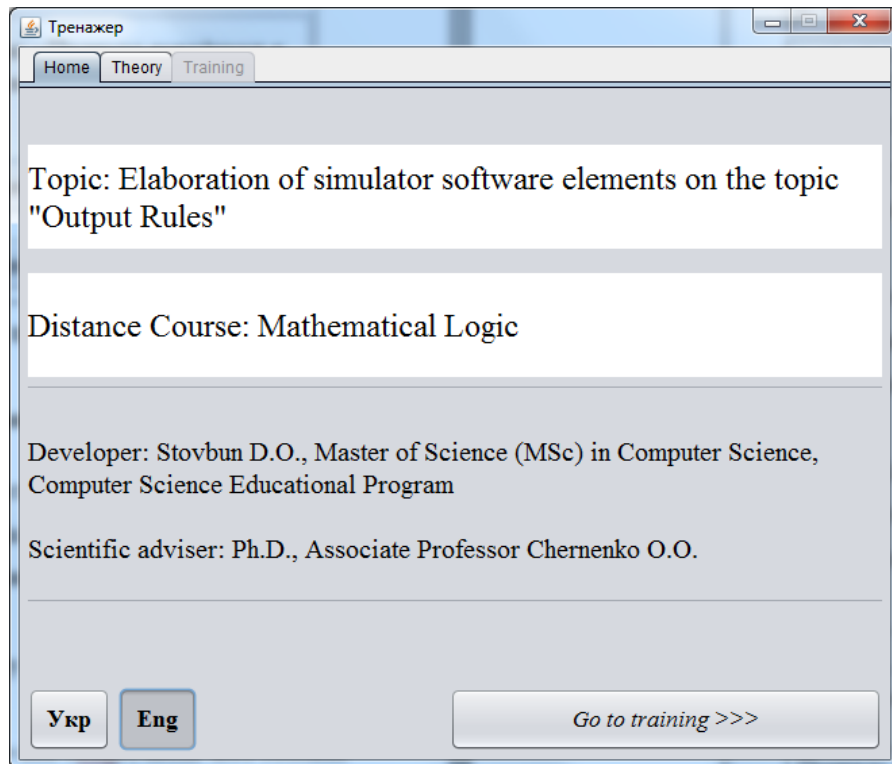


Рисунок 4.9 – Головна сторінка тренажеру англійською мовою

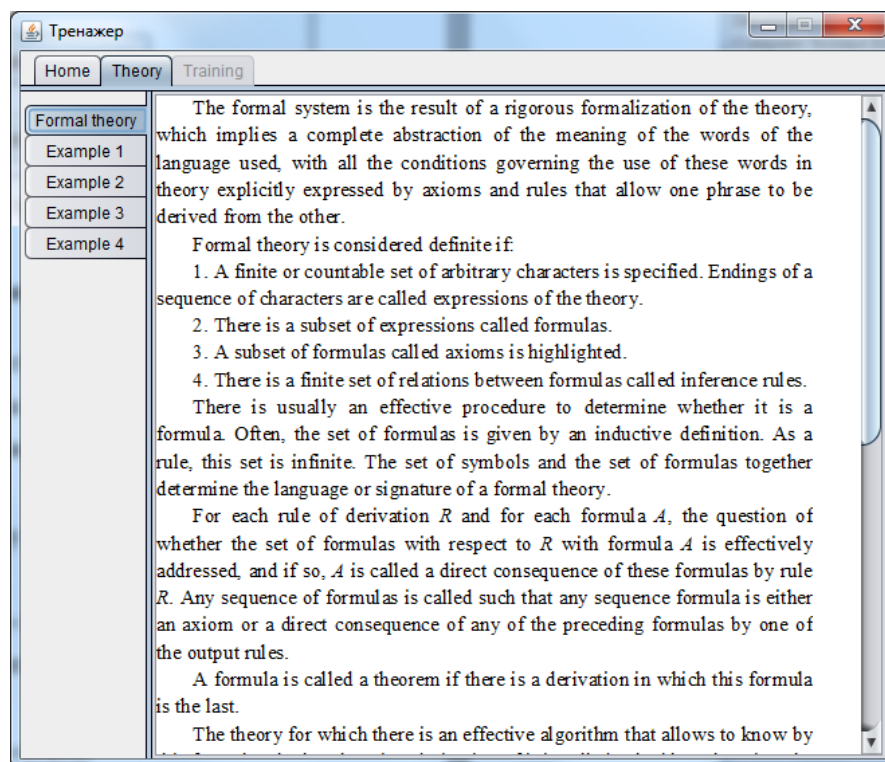


Рисунок 4.10 – Теоретичний матеріал англійською мовою

Спочатку пропонується пройти блок питань «Правила виведення в численні висловлювань». Перше завдання потребує встановлення відповідності між правилами виведення та відповідними їм тавтологіями (рис. 4.4).

Після цього слід натиснути кнопку «Продовжити>>». Якщо встановити відповідність неправильно, то виводиться повідомлення про помилку (рис. 4.11), в іншому разі – наступне завдання.

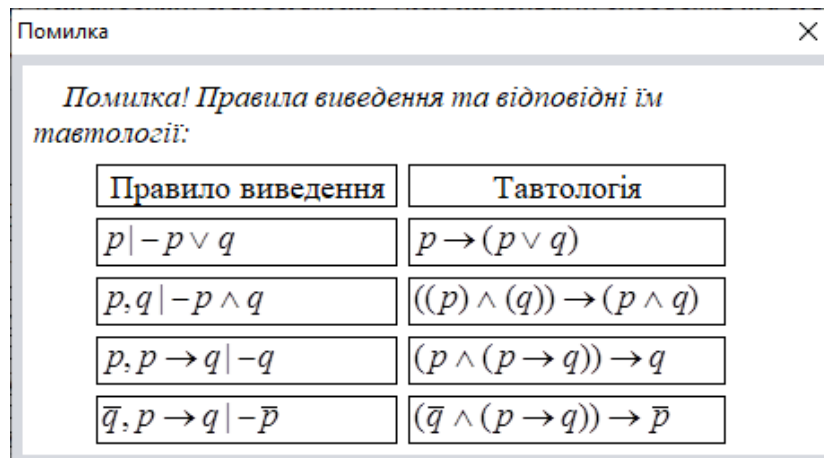


Рисунок 4.11 – Повідомлення про помилку на перше завдання блоку
«Правила виведення в численні висловлювань»

Наступний крок аналогічний, а вже на третьому необхідно встановити відповідність між правилами та їх назвами (рис. 4.12).

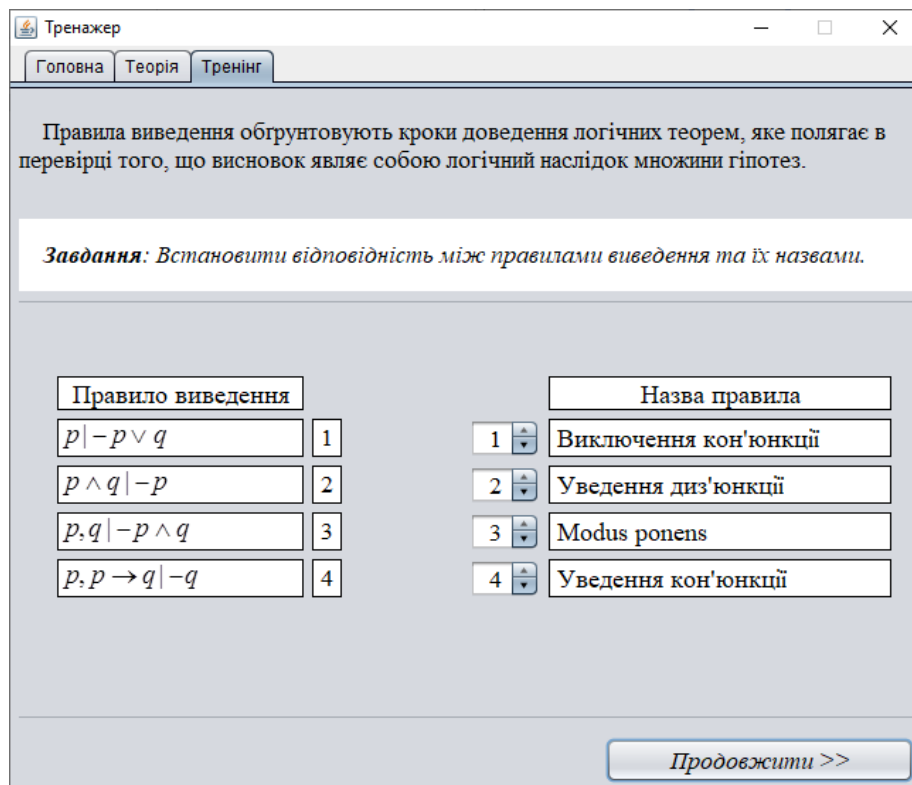


Рисунок 4.12 – Встановлення відповідності на третьому завданні блоку
«Правила виведення в численні висловлювань»

При завершенні блоку «Правила виведення в численні висловлювань» відображається відповідне повідомлення та пропонується перейти до розв’язання прикладів або завершити роботу (рис. 4.8).

Якщо натиснути кнопку «Перейти до розв’язання прикладів», то відбудеться перехід до першого завдання блоку «Застосування правил виведення в численні висловлювань» (рис. 4.5). Якщо натиснути «Завершити роботу тренажеру» – програма закриється.

Якщо вибрати неправильну відповідь, то виведеться повідомлення про помилку (рис. 4.13). Інакше – наступне завдання.

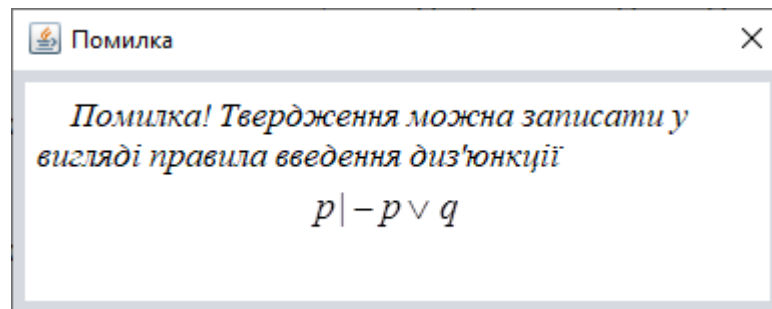


Рисунок 4.13 – Повідомлення про помилку на перше завдання блоку «Застосування правил виведення в численні висловлювань»

Аналогічним чином виконуються й інші завдання.

В завданнях на встановлення послідовності потрібно заповнити поля (рис. 4.6, 4.14).

При завершенні блоку «Застосування правил виведення в численні висловлювань» відображається відповідне повідомлення та пропонується завершити роботу або пройти тренажер знову (рис. 4.15).

Тренажер

Головна Теорія Тренінг

Довести, що з гіпотез «Якщо ти надішлеш мені повідомлення електронною поштою, то я закінчу писати програму», «Якщо ти не надішлеш мені повідомлення електронною поштою, то я рано піду спати» та «Якщо я рано піду спати, то прокинуся бадьорим» випливає висновок «Якщо я не закінчу писати програму, то я прокинуся бадьорим».

Нехай p : «Ти надішлеш мені повідомлення електронною поштою», q : «Я закінчу писати програму», r : «Я рано піду спати», s : «Я прокинуся бадьорим».

Завдання: Гіпотези можна записати у вигляді

$p \rightarrow q, \bar{p} \rightarrow r, r \rightarrow s$. Потрібно обґрунтувати висновок $\bar{q} \rightarrow s$.

Встановити послідовність отримання висновку із заданої множини гіпотез.

<input type="checkbox"/> $\bar{p} \rightarrow r$ – гіпотеза	<input type="checkbox"/> $r \rightarrow s$ – гіпотеза
<input type="checkbox"/> $p \rightarrow q$ – гіпотеза	<input type="checkbox"/> $\bar{q} \rightarrow s$ – гіпотетичний силогізм до 4 та 5
<input type="checkbox"/> $\bar{q} \rightarrow \bar{p}$ – контрапозиція	<input type="checkbox"/> $\bar{q} \rightarrow r$ – гіпотетичний силогізм до 2 та 3

Продовжити >>

Рисунок 4.14 – Шістнадцяте завдання блоку «Застосування правил виведення в численні висловлювань»

Вітаємо! Ви ознайомилися з темою «Застосування правил виведення в численні висловлювань».

Завершити роботу тренажеру

Перейти до початку й пройти тренажер знову

Рисунок 4.15 – Повідомлення про завершення блоку «Застосування правил виведення в численні висловлювань»

Якщо натиснути кнопку «Завершити роботу тренажеру», то програма закриється. Якщо натиснути «Перейти до початку й пройти тренажер знову» – відбудеться перехід до першого завдання блоку «Правила виведення в численні висловлювань».

ВИСНОВКИ

Результатом виконання роботи є програмний продукт, що реалізує тренажер з теми «Правила виведення» дистанційного навчального курсу «Математична логіка».

Основні результати роботи:

- розглянуто тренажери схожої тематики;
- розглянуто тренажери математичних дисциплін з інших тематик;
- визначено позитивні і негативні аспекти оглянутих тренажерів;
- описано актуальність теми роботи;
- розглянуто теоретичні відомості про правила виведення в численні висловлювань;
- розглянуто специфіку застосування правил виведення в численні висловлювань;
- розроблено алгоритм тренажеру, що дозволить закріпити знання з теми «Правила виведення»;
- розроблено навчальний тренажер з даної теми.

Всі розглянуті тренажери були з дисципліни «Методи оптимізації та дослідження операцій», а саме: тренажер з теми «Угорський метод в задачі про призначення», з теми «Графічний метод розв'язування задач лінійного програмування», з теми «Двоїстість в лінійному програмуванні».

В алгоритмі покроково описано всі дії при розв'язанні прикладів, де необхідно застосувати правила виведення в численні висловлювань.

Також реалізовано два блоки прикладів. Перший блок містить завдання з встановлення відповідності між правилами виведення, їх тавтологіями і назвами. У другому описано приклади їх застосування в численні висловлювань.

Після цього було розроблено блок-схему алгоритму.

Так як при реалізації тренажера використано середовище візуальної розробки програм NetBeans IDE та об'єктно-орієнтована мова програмування Java, то спочатку описано причини їх вибору.

Розроблено графічне представлення тренажеру. Панель Main має вкладки:

- Start – стартова сторінка;
- Theory – теоретичний матеріал, поділений на вкладки;
- Training – тренінг, містить дві панелі.

Повідомлення про помилку знаходяться на панелях Ex1Error, Ex2Error, Ex2Error9, Ex2Error16, а про завершення виконання – End.

Для відображення наступного кроку алгоритму або повідомлення про виконання було створено функцію `public void example1Next()` для першого блоку питань і `public void example2Next()` для другого. Якщо поточний крок не останній, то виводиться наступне завдання, інакше – повідомлення про завершення.

Аналогічно створено функції `public void example1Error(int step)` і `public void example2Error(int step)` для виведення помилки.

Щоб всі кнопки працювали потрібним чином на кожну з них було написано відповідну подію.

В тренажері реалізовано можливість переключення між українською та англійською мовами. На кожну кнопку зміни актуальної мови написано відповідну подію.

Останнім етапом проведено опис програми та наведено інструкцію користувача.

Використання програми-тренажера як компонента навчального інформаційного середовища дає можливість не тільки підійти до розв'язання проблеми ефективної інтеграції самостійної роботи в навчальний процес, але також дозволяє без додаткових витрат аудиторного часу заповнити прогалини в знаннях студентів та забезпечити

продуктивний перехід до вивчення основного розділу наукового стилю мови.

Результати роботи були опубліковані в матеріалах науково-практичного семінару «Комп'ютерні науки і прикладна математика (КНіПМ-2019)» та у Збірнику наукових статей магістрів Вищого навчального закладу Укоопспілки «Полтавський університет економіки і торгівлі».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ємець О. О. Методичні рекомендації до виконання дипломної роботи для студентів ступеня магістра спеціальності 122 «Комп'ютерні науки» / О.О.(Олег) Ємець. – Полтава : РВВ ПУЕТ, 2018. – 35 с.
2. Стовбун Д.О. Елементи програмного забезпечення для тренажера з теми «Правила виведення» дистанційного навчального курсу «Математична логіка» / Д.О. Стовбун // Комп'ютерні науки і прикладна математика (КНіПМ-2019): матеріали науково-практичного семінару. Випуск 3 / За ред. Ємця О. О. – Полтава: Кафедра ММСІ ПУЕТ, 2019. – 83 с. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/7042>
3. Самовик С.М. Розробка алгоритму та програмного забезпечення тренажера з теми "Угорський метод в задачі про призначення" дистанційного навчального курсу "Методи оптимізації та дослідження операцій" / С.М. Самовик // Інформатика та системні науки (ІСН-2014) : матеріали V Всеукр.-наук.-практ. конф., (м. Полтава, 13–15 березня 2014 р.). – Полтава: ПУЕТ, 2014. – С. 273-274. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/2840>
4. Самовик С.М. Програмна реалізація угорського методу розв'язування задачі про призначення / С.М. Самовик // Інформатика та системні науки (ІСН-2013) : матеріали IV Всеукр.-наук.-практ. конф., (м. Полтава, 21–23 берез. 2013 р.). – Полтава: ПУЕТ, 2013. – С. 267-268. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/1641>
5. Дистанційний курс «Методи оптимізації та дослідження операцій (Частина 2) (2019) // Головний центр дистанційного навчання вищого навчального закладу УКООПСПЛКІ «Полтавський університет економіки і торгівлі»». – Режим доступу: <http://www2.el.puet.edu.ua/st/course/view.php?id=1287>
6. Крикля М.П. Розробка алгоритму тренажеру з теми "Графічний метод розв'язування задач лінійного програмування"/ М.П. Крикля //

Інформатика та системні науки (ІСН-2014) : матеріали V Всеукр.-наук.-практ. конф., (м. Полтава, 13–15 березня 2014 р.). – Полтава: ПУЕТ, 2014. – С. 163-165. – Режим доступу:

<http://dspace.puet.edu.ua/handle/123456789/2835>

7. Дистанційний курс «Методи оптимізації та дослідження операцій (Частина 1) (2019) // Головний центр дистанційного навчання вищого навчального закладу УКООПСПЛКІ «Полтавський університет економіки і торгівлі»». – Режим доступу:

<http://www2.el.puet.edu.ua/st/course/view.php?id=1211>

8. Душинська А. В. Розробка програми-тренажера з теми «Двоїстість в лінійному програмуванні» дистанційного навчального курсу «Методи оптимізації та дослідження операцій» / А. В. Душинська // Інформатика та системні науки (ІСН-2015): матеріали VI Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 19–21 берез. 2015 р.). – Полтава: ПУЕТ, 2015. – Режим доступу:

<http://dspace.puet.edu.ua/handle/123456789/2498>

9. Нікольський Ю.В. Дискретна математика / Ю.В. Нікольський, В.В. Пасічник, Ю.М. Щербина. – К.: Видавнича група BVH, 2007. – 368 с.

10. Таран Т.А. Основи дискретной математики / Т.А. Таран. – К.: Просвіта, 2003. – 288 с.

11. Жук П. Ф. Математична логіка та теорія алгоритмів : практикум / уклад.: / П. Ф. Жук – К. : НАУ, 2014. – 21 с.

12. IDE NetBeans - более удобный и быстрый способ написания кода: NetBeans IDE Features [Електронний ресурс]. – Режим доступу:

https://netbeans.org/features/index_ru.html

13. Можливості та технології Java: «Студвей» [Електронний ресурс]. – Режим доступу: <https://studway.com.ua/code-part-6/>

14. Шилдт Герберт. Java. Полное руководство, 8-е изд. : Пер. с англ. / Герберт Шилдт. – М. : ООО «И.Д. Вильямс», 2012. – 1104 с.

15. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.